MASTER EN SCIENCES INFORMATIQUES

MetaPIGA: Optimisation et évaluation des performances d'un programme informatique pour l'inférence phylogénétique utilisant un algorithme génétique métapopulationnel.

Gribaumont, Chantal

Award date:
2009

Université Libre de Bruxelles
Facultés des Sciences

Mémoire de fin d'études présenté en vue de l'obtention du diplôme de
Master en Bioinformatique et Modélisation (juin 2009)

MetaPIGA : Optimisation et évaluation des performances d'un programme informatique pour l'inférence phylogénétique utilisant un algorithme génétique métapopulationnel

Chantal GRIBAUMONT

Promoteurs : Michel C. MILINKOVITCH (Université de Genève)
Timoteo CARLETTI (Facultés Universitaires Notre-Dame de la Paix, Namur)
Albert GOLDBETER (Université Libre de Bruxelles)

Année académique 2008-2009

Remerciements

Lors d'une de nos premières discussions concernant les mémoires de fins d'études avec le Professeur J. Van Helden (alors secrétaire du jury du Master en Bioinformatique et Modélisation), ce dernier m'a dit : « Vous, vous êtes venue en bioinformatique pour faire de la phylogénie ».

Il serait sans doute plus juste de dire que c'est la découverte des méthodes d'inférence phylogénétique qui m'ont amenée à étudier la bioinformatique. Il est cependant vrai que l'étude de l'évolution a toujours été mon but, depuis mes débuts en Bachelier en Sciences Biologiques, voire même depuis qu'on m'en a parlé pour la première fois, pendant mes études secondaires.

Il était donc tout naturel pour moi de demander au Professeur M. C. Milinkovitch d'être promoteur de mon mémoire. Il était par contre bien moins naturel pour lui d'accepter, étant donné qu'il venait d'obtenir un poste à l'Université de Genève, et qu'il avait certainement assez à faire avec la (re)création de son laboratoire en Suisse. Je voudrais donc le remercier tout particulièrement, pour m'avoir permis de travailler avec lui.

Merci également au Professeur T. Carletti qui, en étant mon promoteur aux Facultés Universitaires Notre-Dame de la Paix à Namur, a facilité les choses, et permis que j'aie un espace de travail aux côtés de Raphaël Helaers.

Merci à Raphaël Helaers, chercheur aux Facultés Universitaires Notre-Dame de la Paix à Namur, qui a écrit le programme MetaPIGA, et a inlassablement répondu à mes questions et rapidement corrigé les bugs qui freinaient mon travail.

Merci enfin à tous les membres du laboratoire de bioinformatique à l'unité de recherche en biologie moléculaire des Facultés Universitaires Notre-Dame de la Paix à Namur, qui m'ont accueillie dans leur bureau.

Résumé

La phylogénie est l'étude des relations évolutives entre espèces (ou gènes, ou populations...). Ces relations sont représentées à l'aide d'arbres phylogénétiques, et la reconstruction de ces arbres à partir de données moléculaires ou morphologiques est une branche importante de la biologie.

Diverses méthodes existent, parmi lesquelles les méthodes de distances, les méthodes utilisant le principe de parcimonie, les méthodes Bayesiennes et les méthodes utilisant le maximum de vraisemblance. Ce sont ces dernières méthodes qui sont utilisées par le programme MetaPIGA, sujet de ce travail.

Les méthodes utilisant le maximum de vraisemblance donnent de très bons résultats, et surpassent les méthodes de distances ou utilisant le principe de parcimonie. Le principe des méthodes utilisant le maximum de vraisemblance est de comparer tous les arbres possibles et de choisir l'arbre ayant la plus haute vraisemblance, c'est-à-dire la plus haute probabilité de produire les données observées (dans le cas de MetaPIGA, il s'agit de séquences ADN alignées). Cependant, le nombre d'arbres possibles augmentant de manière exponentielle avec le nombre de taxa considérés, il n'est pas possible de comparer les vraisemblances de tous les arbres dès lors que l'on travaille avec des phlyogénies de plus de quelques taxa. Par conséquent, on utilise des heuristiques, qui sont des algorithmes permettant de trouver une bonne solution à des problèmes irrésolubles en un temps raisonnable, mais ne peuvent garantir de trouver la solution optimale.

Quatre heuristiques différentes peuvent être utilisée par MetaPIGA : le Hill-Climbing, le recuit simulé, un algorithme génétique et un algorithme génétique métapopulationnel.

Ce travail a consisté à tester différents paramètres pour chaque heuristique, afin de choisir ceux qui donnent les meilleurs résultats. Cependant, les tests n'ont été effectués que sur un set de données simulées, comprenant seulement 20 taxa, et chaque paramètre n'a été testé que 50 fois. La pincipale conclusion que l'on peut tirer des résultats obtenus est que les heuristiques risquent fort de se comporter différemment sur des données comprenant un plus grand nombre de taxa. Or MetaPIGA, et en particulier l'algorithme génétique métapopulationnel, a été créé spécifiquement pour pouvoir gérer des données de grandes tailles.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Chapitre 1

Introduction

La phylogénie - du grec $\phi\upsilon\lambda o\nu$ *phylon* (tribu, race) et $\gamma\epsilon\nu\nu\omega$ *genno* (donner naissance, créer) - est l'étude de l'histoire et des relations évolutives de groupes d'espèces, d'individus ou autres « unités taxonomiques ».

Depuis Darwin et la théorie de l'évolution (Darwin, 1859), selon laquelle tous les organismes partagent une origine commune et ont divergé avec le temps, les biologistes ont tenté de reconstruire l'histoire évolutive de la vie.

Au-delà de l'intérêt purement fondamental de cette question, la phylogénie est utile de bien des manières. En génétique de la conservation par exemple, connaître l'histoire évolutive des espèces, voire simplement être capable de connaître les relations entre plusieurs individus peut parfois aider à sauver une espèce. C'est également la phylogénie qui permet aujourd'hui d'affirmer que le premier être humain est apparu en Afrique. Cela peut aussi servir entre autres à comprendre les phénomènes de disparitions et gains de gènes, dater des événements de spéciation, détecter une accélération de l'évolution, observer la cospéciation d'un parasite et de son hôte ou retracer l'évolution d'un virus à travers ses différentes souches.

Aujourd'hui, les données moléculaires sont de plus en plus nombreuses, et les phylogénies à reconstruire de plus en plus grandes. Or, la reconstruction de l'histoire évolutive des gènes, espèces ou individus n'est pas une tâche triviale, loin s'en faut. Par ailleurs, il semblerait que les arbre phylogénétiques soient plus fiables lorsqu'inférés grâce à de nombreux taxa (Hillis, 1996). Couplé au fait que les méthodes donnant les meilleurs résultats (méthodes utilisant le maximum de vraisemblance et méthodes Bayesiennes) sont aussi les plus longues et les plus computationnellement intensives, cela provoque une grosse demande pour des programmes efficaces et rapides, capables de reconstruire un arbre phylogénétique pour de grands sets de données en un temps raisonnable.

MetaPIGA est un programme d'inférence phylogénétique (Lemmon and Milinkovitch, 2002), encore en cours de développement mais déjà fonctionnel, qui tente de répondre à ce besoin. Il est basé sur le critère du maximum de vraisemblance et utilise notamment une variante des algorithmes génétiques.

1.1 But du travail

Le but de ce travail était d'optimiser autant que possible les performances de MetaPIGA, en testant un certain nombre de ses paramètres et en comparant les résultats obtenus pour les différents choix de paramétrage.

MetaPIGA implémente quatre algorithmes différents pour la recherche d'arbres phylogénétiques, et chacun de ces algorithmes a été partiellement optimisé. Les quatre méthodes ont ensuite été comparées afin de voir laquelle produit les meilleurs résultats.

1.2 Plan du travail

La suite de ce document est organisée comme suit : le chapitre 2 explique les principes de l'inférence phylogénétique, ainsi que les principales méthodes utilisées. Le chapitre 3 décrit de manière assez exhaustive le programme MetaPIGA sur lequel est basé ce travail, tandis que le chapitre 4 décrit succintement quelques autres programmes d'inférence phylogénétique parmi les plus utilisés. Les chapitres 5 et 6 expliquent les analyses effectuées sur MetaPIGA et donnent les résultats obtenus. Enfin, le chapitre 7 conclut ce travail et donne quelques pistes pour de futures analyses.

Chapitre 2

Inférence phylogénétique

Avant l'apport de la biologie moléculaire, les arbres phylogénétiques étaient construits sur base de caractères morphologiques, anatomiques et physiologiques.

Dorénavant, grâce notamment aux techniques de séquençage de l'ADN, les arbres phylogénétiques sont construits à partir de séquences génétiques et protéiques. Ce type de données présente plusieurs avantages : pour commencer, les caractères moléculaires sont moins subjectifs et ambigus que les caractères morphologiques. Ensuite, ils permettent d'inférer les relations évolutives entre des organismes très éloignés. De plus, ces caractères évoluent généralement de manière plus régulière et homogène que les caractères morphologiques ou physiologiques. Et enfin, les données moléculaires peuvent être facilement traitées de manière quantitative.

Ce chapitre présente les grands principes de l'inférence phylogénétique, ainsi que les méthodes les plus utilisées.

Ces méthodes sont présentées en faisant l'hypothèse que les données utilisées pour construire les arbres phylogénétiques sont des séquences ADN alignées, mais elles sont bien entendu applicables à d'autres types de données, telles des séquences ARN, protéiques...

2.1 Représentation d'un arbre phylogénétique

Un arbre phylogénétique est représenté par un graphe, composé de branches et de noeuds. Un noeud représente une unité taxonomique ou taxon (une espèce, un individu, un gène, une population...), tandis que les branches montrent les relations entre ces unités taxonomiques (exemple figure 2.1). La topologie d'un arbre désigne la façon dont les branches relient les noeuds, et la longueur d'une branche donne une mesure de la distance ou de la similarité entre les unités taxonomiques qu'elle relie.

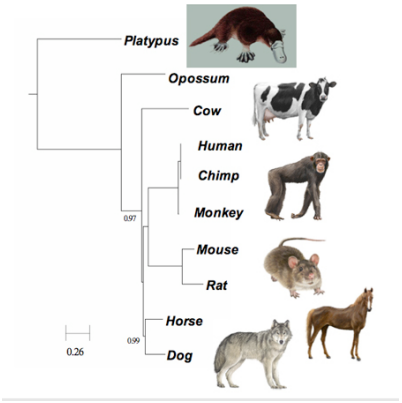
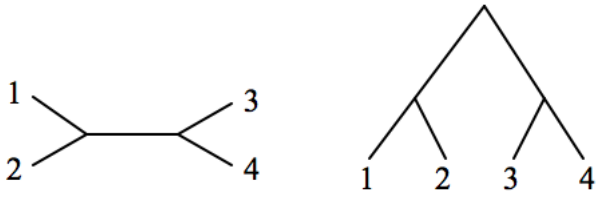


FIG. 2.1 – Exemple d'arbre phylogénétique pour les mammifères
(http ://www.universityofcalifornia.edu/news/article/19408)

Un noeud est dit dichotomique lorsqu'il est le point de jonction de trois branches, et polytomique lorsqu'il y a plus de trois branches. Par extension, un arbre dichotomique est un arbre dont tous les noeuds sont dichotomiques et un arbre polytomique, un arbre dont au moins un noeud est polytomique. L'évolution étant un phénomène bifurcatif, pour être complètement résolu un arbre doit être dichotomique.

On distingue les arbres racinés des arbres non-racinés (figure 2.2). Lorsqu'un arbre est raciné, sa racine représente l'ancêtre commun le plus récent de toutes les unités taxonomiques de l'arbre. Un arbre raciné est donc dirigé, et représente ce qu'on peut appeler un « chemin évolutif », de l'ancêtre commun aux unités taxonomiques actuelles, tandis qu'un arbre non raciné ne donne pas d'indication de direction, mais seulement les relations entre les unités taxonomiques.

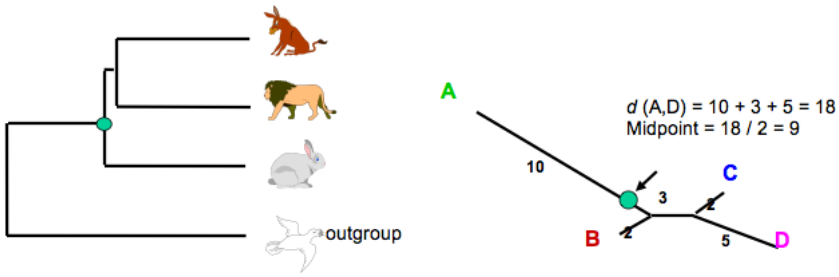


(a) Arbre non raciné (b) Arbre raciné

FIG. 2.2 – Exemple d'arbres génétiques pour quatre taxa. (a) Arbre non raciné. (b) Arbre raciné.

Pour enraciner un arbre, il existe deux méthodes (figure 2.3).

- Soit on utilise un « outgroup », c'est-à-dire un taxa dont on sait qu'il est extérieur aux unités taxonomiques considérés, donc que le plus proche ancêtre commun de cet outgroup et des unités taxonomiques considérées est antérieur au plus proche ancêtre commun de ces unités taxonomiques. Cela nécessite d'avoir une connaissance a priori des relations taxonomiques.
- Soit on fixe la racine au « point moyen », c'est-à-dire entre les deux taxa les plus éloignés. Cela suppose que la vitesse d'évolution est la même sur toutes les branches, ce qui n'est pas une hypothèse très réaliste.



(a) Enracinement par outgroup (b) Enracinement par point moyen

FIG. 2.3 – (Stewart, 2000) Les deux méthodes d'enracinement des arbres phylogénétiques. (a) Enracinement par outgroup : arbre phylogénétique pour des mammifères enraciné grâce à un oiseau. (b) Enracinement par point moyen.

2.2 Modèles de mutation de l'ADN

Toutes les méthodes d'inférences phylogénétiques à partir de séquences ADN nécessitent une estimation du nombre de substitutions ayant eu lieu d'une séquence à l'autre. Si l'on estime directe-

ment, par simple comptage des différences entre les séquences, ce nombre de substitutions, on risque fort de sous-estimer celui-ci, en particulier quand le temps de divergence entre les deux séquences est long.

En effet, lorsque l'on dénombre simplement les sites ayant une base différente dans chacune des séquences, on considère qu'à chacun de ces sites, une et une seule substitution a eu lieu. En réalité, *au minimum* une substitution a eu lieu (Figure 2.4). Par ailleurs, il est aussi possible que certains sites ait subi plusieurs substitutions, dont la dernière a réuniformisé les sites des deux séquences. Ces substitutions sont dès lors invisibles.

Plusieurs modèles ont été développés afin de tenir compte de ce problème, dont cinq, parmi les plus utilisés, sont présentés ci-dessous : modèles de Jukes-Cantor (JC) (Jukes and Cantor, 1969), Kimura-2-paramètres (K2P) (Kimura, 1980, 1981), Hasegawa-Kishino-Yano (1985) (HKY85) (Hasegawa *et al.*, 1985), Tamura-Nei (1993) (TN93) (Tamura and Nei, 1993) et « General Time Reversible (GTR) »(Tavaré, 1986).

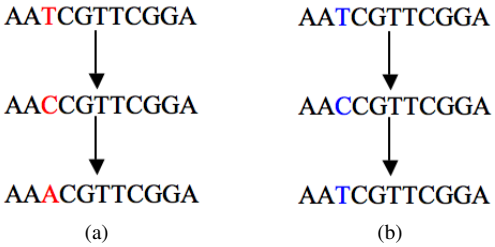


FIG. 2.4 – (a) Une seule mutation est visible, pourtant deux substitutions ont eu lieu. (b) Aucune mutation n'est visible malgré que deux substitutions aient eu lieu.

2.2.1 Modèle de Jukes-Cantor

Dans le modèle de JC (Jukes and Cantor, 1969), on considère que toutes les substitutions ont une même probabilité, et que la fréquences des bases ADN est $\pi_A = \pi_T = \pi_C = \pi_G = \frac{1}{4}$.

Sachant qu'il existe quatre bases ADN différentes, trois substitutions sont possibles pour chacune d'entre-elles, ce qui fait 12 substitutions possibles (figure 2.5). On peut donc écrire la matrice de taux de substitution instantané suivante :

$$R = \begin{pmatrix} & A & C & G & T \\ A & \cdot & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} \\ C & \frac{\alpha}{4} & \cdot & \frac{\alpha}{4} & \frac{\alpha}{4} \\ G & \frac{\alpha}{4} & \frac{\alpha}{4} & \cdot & \frac{\alpha}{4} \\ T & \frac{\alpha}{4} & \frac{\alpha}{4} & \frac{\alpha}{4} & \cdot \end{pmatrix}$$

Avec $r_{ij} = \alpha \times \pi_i$: taux de substitution instantané de i par $j \; \forall \, i, j \in \{A, C, G, T\}$ et $i \neq j$.
Les points « $\cdot$ » sont des valeurs telles que la somme des lignes et des colonnes vaut 0.

Donc, si au temps t_0, la base en position x est i, la probabilité que ce site soit toujours i au temps t_1 est :

$$P_{i(t_1)} = 1 - 3\alpha\prime \qquad \text{avec } \alpha\prime = \frac{\alpha}{4}$$

Et la probabilité d'avoir toujours i au temps $t + 1$ est de :

$$P_{i(t+1)} = P_{i(t)} \cdot (1 - 3\alpha\prime) + (1 - P_{i(t)}) \cdot \alpha\prime$$

Que l'on peut réécrire comme suit :

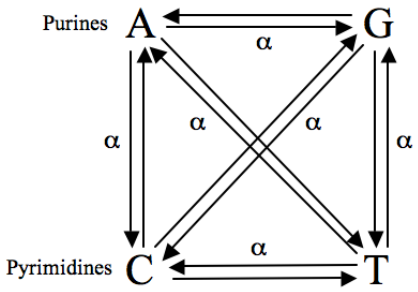


FIG. 2.5 – Substitutions possibles entre les quatre bases ADN, modèle de JC (α est le taux de substitution). A : adénine, T : thymine, C : cytosine et G : guanine.

$$P_{i(t+1)} - P_{i(t)} = -3\alpha\prime P_{i(t)} + \alpha\prime - \alpha\prime P_{i(t)}$$

Ou :

$$\Delta P_{i(t)} = -4\alpha\prime P_{i(t)} + \alpha\prime$$

Et si l'on passe à un processus continu :

$$\frac{dp_{i(t)}}{dt} = -4\alpha\prime P_{i(t)} + \alpha\prime \tag{2.1}$$

Après intégration de l'équation 2.1, on obtient :

$$P_{i(t)} = \frac{1}{4} + \left[P_{i(0)} - \frac{1}{4}\right] e^{-4\alpha\prime t} \tag{2.2}$$

Ainsi, si l'on commence avec i au site x, la probabilité qu'on ait toujours la base i au temps t est :

$$P_{ii(t)} = \frac{1}{4} + \frac{3}{4}e^{-4\alpha\prime t} \qquad (P_{i(0)} = 1) \tag{2.3}$$

Et la probabilité qu'on ait la base j $(j \neq i)$ au temps t est :

$$P_{ij(t)} = \frac{1}{4} - \frac{1}{4}e^{-4\alpha\prime t} \qquad (P_{j(0)} = 0) \tag{2.4}$$

Les équations 2.3 et 2.4 tendent toutes deux vers $0,25$ quand t tend vers ∞.

Grâce à ces deux équations, il est possible de calculer la divergence entre deux séquences s'étant séparées il y a un temps t. En effet, pendant ce temps t, les séquences ont évolué chacune de leur côté, menant à un temps de divergence de $2t$.

La probabilité qu'un site x comporte la même base i dans les deux séquences (probabilité d'identité) est donc :

$$I_t = P_{ii(2t)} = \frac{1}{4} + \frac{3}{4}e^{-4\alpha\prime 2t} = \frac{1}{4} + \frac{3}{4}e^{-8\alpha\prime t} \tag{2.5}$$

Et la probabilité qu'un site x comporte des bases différentes dans les deux séquences est bien entendu de :

$$D_t = 1 - I_t = \frac{3}{4}\left(1 - e^{-8\alpha\prime t}\right) \tag{2.6}$$

Le but du modèle de JC étant d'estimer le nombre de substitutions ayant réellement eu lieu lorsque l'on observe D différences entre deux séquences, on réorganise l'équation 2.6, pour obtenir :

$$\boxed{K = -\tfrac{3}{4}\ln\left(1 - \tfrac{4D}{3}\right)} \tag{2.7}$$

Avec $K = 2 \times 3\alpha\prime \times t$: le vrai nombre de substitutions par site
(le long de deux branches de l'arbre phlogénétique)
D : le nombre de substitutions observées

Comme on peut le voir sur la figure 2.6, lorsque l'on observe peu de différences entre les deux séquences, la relation entre le nombre estimé et le nombre observé de substitutions est presque linéaire. Mais lorsque la divergence augmente, le nombre de substitutions estimé explose. En particulier, dès que l'on observe 75% de divergence ou plus, le nombre de substitutions réel est virtuellement infini puisque deux séquences générées au hasard auront 25% de sites identiques (donc 75% de divergence observée).

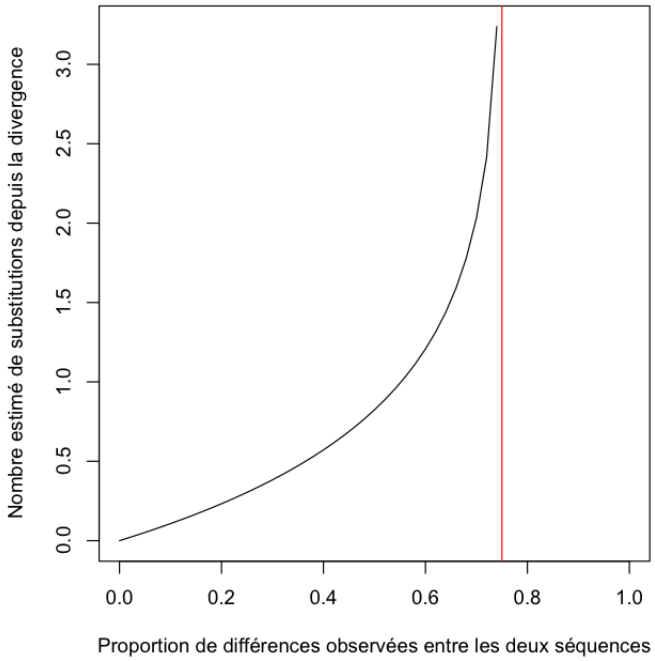


FIG. 2.6 – Estimation du nombre de substitutions entre deux séquences selon le modèle de JC. Ligne rouge : 75% de divergence observée.

Le modèle de JC est le modèle de substitution le plus simple. C'est aussi celui qui fait le plus d'hypothèses a priori et, par conséquent, le moins réaliste.

2.2.2 Modèle Kimura-2-paramètres

Le modèle K2P (Kimura, 1980, 1981) différencie les substitutions transformant une purine en pyrimidine ou inversément (transversions) des substitutions entre purines ou pyrimidines (transitions) en leur attribuant une probabilité différente (figure 2.7). Les fréquences nucléotidiques sont toujours considérées comme égales ($\pi_A = \pi_C = \pi_G = \pi_T$).

On a donc comme matrice de taux de substitution instantané :

$$R = \begin{pmatrix} & A & C & G & T \\ A & \cdot & \frac{\beta}{4} & \frac{\alpha}{4} & \frac{\beta}{4} \\ C & \frac{\beta}{4} & \cdot & \frac{\beta}{4} & \frac{\alpha}{4} \\ G & \frac{\alpha}{4} & \frac{\beta}{4} & \cdot & \frac{\beta}{4} \\ T & \frac{\beta}{4} & \frac{\alpha}{4} & \frac{\beta}{4} & \cdot \end{pmatrix}$$

Avec α : taux de transition
β : taux de transversion
$r_{ij} = \pi_i \times$taux de substitution (α ou β) : taux de substitution instantané de i par $j \; \forall \, i, j \in \{A, C, G, T\}$ et $i \neq j$

Les points « · » sont des valeurs telles que la somme des lignes et des colonnes vaut 0.

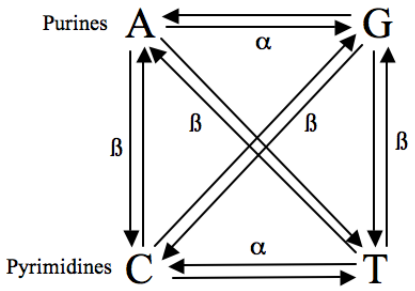


FIG. 2.7 – Substitutions possibles entre les quatre bases ADN, modèle K2P (α est le taux de transition et β le taux de transversion).

2.2.3 Modèle Hasegawa-Kishino-Yano (1985)

Ce modèle (Hasegawa *et al.*, 1985) distingue également les transitions des transversions, mais ne fait plus l'hypothèse des fréquences nucléotidiques égales ($\pi_A \neq \pi_C \neq \pi_G \neq \pi_T$).

$$R = \begin{pmatrix} & A & C & G & T \\ A & \cdot & \beta\pi_C & \alpha\pi_G & \beta\pi_T \\ C & \beta\pi_A & \cdot & \beta\pi_G & \alpha\pi_T \\ G & \alpha\pi_A & \beta\pi_C & \cdot & \beta\pi_T \\ T & \beta\pi_A & \alpha\pi_C & \beta\pi_G & \cdot \end{pmatrix}$$

Avec α : taux de transition
β : taux de transversion
π_i : fréquence de la base $i \quad \forall \, i \in \{A, C, G, T\}$
$r_{ij} = \pi_i \times$taux de substitution (α ou β) : taux de substitution instantané de i par $j \; \forall \, i, j \in \{A, C, G, T\}$ et $i \neq j$

Les points « · » sont des valeurs telles que la somme des lignes et des colonnes vaut 0.

2.2.4 Modèle Tamura-Nei (1993)

Dans le modèle TN93 (Tamura and Nei, 1993), on considère que les transitions peuvent être de deux types. : $A \leftrightarrow G$ est différent de $C \leftrightarrow T$.

$$R = \begin{pmatrix} & A & C & G & T \\ A & \cdot & \alpha_1\pi_C & \alpha_3\pi_G & \alpha_1\pi_T \\ C & \alpha_1\pi_A & \cdot & \alpha_1\pi_G & \alpha_2\pi_T \\ G & \alpha_3\pi_A & \alpha_1\pi_C & \cdot & \alpha_1\pi_T \\ T & \alpha_1\pi_A & \alpha_2\pi_C & \alpha_1\pi_G & \cdot \end{pmatrix}$$

Avec α_1 : taux de transversion
α_2 : taux de transition $C \leftrightarrow T$
α_3 : taux de transition $A \leftrightarrow G$
π_i : fréquence de la base $i \quad \forall \, i \in \{A, C, G, T\}$
$r_{ij} = \alpha_s \times \pi_i$: taux de substitution instantané de i par $j \; \forall \, i, j \in \{A, C, G, T\}$, $i \neq j$ et $s \in \{1, 2, 3\}$

Les points « · » sont des valeurs telles que la somme des lignes et des colonnes vaut 0.

2.2.5 Modèle « General Time Reversible »

Comme son nom l'indique, le modèle GTR (Tavaré, 1986) est le plus général. La seule hypothèse de ce modèle est que $P(i \to j) = P(j \to i)$. Chaque substitution a son propre taux (il y a donc six taux de substitutions différents) et les fréquences nucléotidiques ne sont pas supposées égales.

$$R = \begin{pmatrix} & A & C & G & T \\ A & \cdot & \alpha_1\pi_C & \alpha_2\pi_G & \alpha_3\pi_T \\ C & \alpha_1\pi_A & \cdot & \alpha_4\pi_G & \alpha_5\pi_T \\ G & \alpha_2\pi_A & \alpha_4\pi_C & \cdot & \alpha_6\pi_T \\ T & \alpha_3\pi_A & \alpha_5\pi_C & \alpha_6\pi_G & \cdot \end{pmatrix}$$

Avec α_s : taux de substitution
π_i : fréquence de la base $i \quad \forall\, i \in \{A, C, G, T\}$
$r_{ij} = \alpha_s \times \pi_i$: taux de substitution instantané de i par j
$\forall\, i, j \in \{A, C, G, T\}, i \neq j$ et $s \in \{1, 2, 3, 4, 5, 6\}$

2.2.6 Hétérogénéité des taux de substitution

Il a été montré que le taux de substitution varie d'un site à l'autre (Uzzell and Corbin, 1971). De plus certains sites semblent ne jamais subir de mutations (Fitch and Margoliash, 1967a).

Cette hétérogénéité peut être modélisée avec un taux d'invariants et/ou une distribution probabiliste des taux.

Taux d'invariants

Ce taux peut être estimé grâce à une méthode de maximum de vraisemblance. Un modèle de substitutions d'ADN permettant à une fraction des sites de ne jamais muter semble s'ajuster mieux aux données qu'un modèle avec un taux de substitution uniforme (Adachi and Hasegawa, 1995).

Distribution des taux de substitution

L'hétérogénéité du taux de substitution peut être modélisé grâce à une distribution continue. La distribution gamma (Γ) est la plus utilisée, en raison de ses propriétés mathématiques (Yang, 1993). Elle permet en effet, en changeant un seul paramètre (α : paramètre de forme), d'avoir des distributions fort différentes (figure 2.8).

A chaque site est donc assigné un taux de substitution tiré de la distribution Γ utilisée.

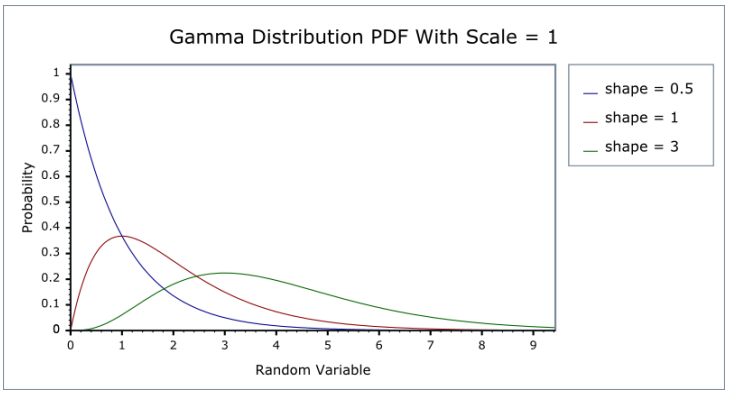


FIG. 2.8 – Densité de probabilité de la distribution gamma pour différents paramètres de forme (source : http ://www.boost.org/doc/html/index.html)

2.3 Méthodes de distance pour l'inférences phylogénétique

Le principe général de ces méthodes (Sokal and Michener, 1958; Cavalli-Sforza and Edwards, 1967; Fitch and Margoliash, 1967b) est de calculer une distance entre chaque paires d'espèces, pour ensuite trouver l'arbre phylogénétique qui prédit le mieux ces distances. La distance entre deux espèces peut être simplement le nombre de différences entre leurs séquences génétiques, ou cette même distance corrigée par un modèle de mutation de l'ADN. Les longueurs de branches de l'arbre phylogénétique construit seront basées sur ces distances, et reflèteront donc non pas uniquement le temps écoulé depuis la bifurcation de la branche, mais surtout le degré d'évolution de cette branche (la vitesse d'évolution n'étant pas nécessairement la même sur toutes les branches de l'arbre).

Deux méthodes de distances sont présentées ci-dessous : la méthode Unweight Pair Group Method with Arithmetic mean (UPGMA) et le Neighbour-Joining (NJ).

2.3.1 Unweight Pair Group Method with Arithmetic mean

Cette méthode (Michener and Sokal, 1957) est en fait un simple clustering hiérarchique. Les distances entre toutes les paires d'espèces sont calculées (éventuellement en utilisant un modèle de substitution de l'ADN), chaque taxon représentant un cluster de taille 1. On considère que les longueurs de branches (donc les distances) respectent le principe de l'horloge moléculaire (Kimura, 1968), c'est-à-dire que la vitesse d'évolution des séquences est constante et est la même sur toutes les branches de l'arbre.

Les deux clusters les plus proches sont joints (formant un noeud interne de l'arbre), et les distances entre ce nouveau cluster et les autres clusters sont recalculées. On continue ainsi à joindre les clusters jusqu'à réunion de tous les taxa dans l'arbre phylogénétique.

Cette méthode est très rapide mais est sujette à d'importants biais. En particulier, si les branches n'ont pas toutes le même taux de substitution, donc ne respectent pas le principe de l'horloge moléculaire, la topologie sera probablement fausse (figure 2.9).

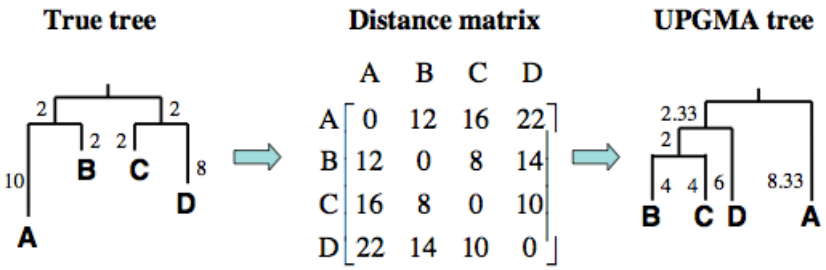


FIG. 2.9 – (Lin, 2008) Arbre phylogénétique à quatre taxa. A gauche, le « vrai » arbre, au milieu, la matrice de distance correspondante. On voit que la méthode d'UPGMA donne un arbre erroné (à droite). En cause : les vitesses d'évolution différentes d'une branche à l'autre.

2.3.2 Neighbour-Joining

L'algorithme du NJ (Saitou and Nei, 1987) autorise les branches de l'arbre phylogénétique à montrer des taux de substitution différents, en prenant en compte non seulement la distance entre chaque paire d'espèces mais aussi leur distance respective aux autres espèces.

On démarre avec un arbre en étoile (un seul noeud interne) et on construit une matrice de distance M de la manière suivante :

$$m_{ij} = d_{ij} - \frac{r_i + r_j}{N - 2} \tag{2.8}$$

Avec d_{ij} : la divergence entre la séquence i et la séquence j
(éventuellement corrigée par un modèle de substitutions de l'ADN)
$r_i = \sum_j d_{ij} \quad \forall\, j \in \{\text{ensemble des séquences considérées}\}$ et $j \neq i$
N : le nombre de séquences considérées

On définit un nouveau noeud interne, parent des séquences ayant le m_{ij} le plus petit. La distance entre ce nouveau noeud et les noeuds joints est calculée comme suit :

$$d_{au} = \frac{d_{ab}}{2} + \frac{r_a - r_b}{2(N-2)} \tag{2.9}$$

$$d_{bu} = d_{ab} - d_{au} \tag{2.10}$$

Avec a et b : les deux noeuds joints
u : le nouveau noeud, résultat du clustering de a et b

Et on recalcule la matrice M (le nouveau noeud interne remplace les deux noeuds joints) :

$$d_{uj} = d_{aj} + d_{bj} - \frac{d_{ab}}{2} \tag{2.11}$$

N vaut maintenant $N - 1$, et on recommence le processus jusqu'à ce que $N = 2$.

Le Neighbourg-Joining est un algorithme rapide, qui donne de bons résultats, en particulier pour de petites phylogénies. De plus, de nombreuses variantes de l'algorithme de départ diminuant encore le temps de calcul existent (Elias and Lagergren, 2008; Simonsen *et al.*, 2008; Sheneman *et al.*, 2006).

2.4 Méthodes utilisant la parcimonie

Lorsque l'on utilise une méthode de distances, toute l'information à propos d'une séquence est résumée en un chiffre. Il y a donc une grosse perte d'informations. La parcimonie est une méthode basée non plus sur les distances mais sur les caractères. L'information relative à chaque site de chaque séquence sera utilisée.

Le principe de parcimonie peut être résumé comme ceci : lorsque l'on a le choix entre plusieurs hypothèses, la plus simple doit être favorisée. Appliqué à la phylogénie, cela signifie que l'arbre phylogénétique expliquant les séquences observées en nécessitant le plus petit nombre d'événements évolutifs (mutation, additions, délétions) est considéré comme le « meilleur » arbre.

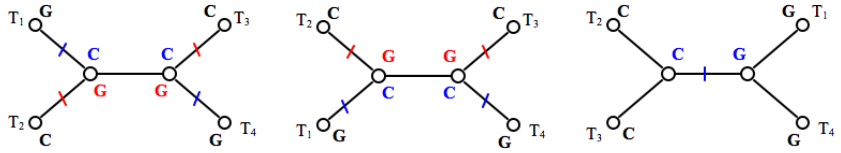
Par exemple, pour construire un arbre phylogénétique à partir des quatre séquences suivantes :

T_1	G	T	T	C	T	T	C	T	C	A	T	T	A	G	A	G	T	T
T_2	C	A	T	C	A	T	C	T	T	A	T	A	A	G	A	G	T	C
T_3	C	T	T	G	T	T	C	C	A	A	T	A	A	G	A	G	T	T
T_4	G	T	T	C	T	A	C	T	A	A	T	T	A	G	A	G	T	A

On considère chaque caractère séparément et on construit tous les arbres possibles. Donc, pour le premier caractère (GCCG), on construit les trois arbres (non racinés) possibles. Pour chacun de ces arbres, on compte le nombre d'événements évolutifs nécessaires pour expliquer les données de ce site (figure 2.10).

On refait ensuite la même chose pour chaque caractère, et on obtient la table 2.1. Au total, la topologie 3 requiert donc moins d'événements évolutifs que les topologies 1 et 2, et sera donc choisie.

Cependant, si cette méthode tient compte de chaque caractère individuellement, elle ne permet de choisir qu'entre différentes topologies. Or, un arbre phylogénétique n'est pas seulement défini par sa topologie, mais également par des longueurs de branches.



(a) Topologie 1 (b) Topologie 2 (c) Topologie 3

FIG. 2.10 – Les trois arbres représentent les trois topologies possibles pour les quatre taxa (T_1, T_2, T_3 et T_4). (a) Il faut minimum deux événement évolutifs pour réconcilier l'arbre et les données pour le premier caractère. Les deux possibilités sont indiquées respectivement en bleu et en rouge, les barres coupant les branches représentent une mutation. (b) Cette topologie est presque la même que la topologie 1. A nouveau, deux événements évolutifs sont nécessaires pour expliquer les données. (c) Pour cette topologie, seul un événement est nécessaire pour réconcilier l'arbre phylogénétique avec les données.

Position	1	2	3	4	5	6	6	8	9	10	11	12	13	14	15	16	17	Somme
Topologie 1	2	1	0	1	1	1	0	1	2	0	2	0	0	0	0	0	2	13
Topologie 2	2	1	0	1	1	1	0	1	2	0	2	0	0	0	0	0	2	13
Topologie 3	1	1	0	1	1	1	0	1	2	0	1	0	0	0	0	0	2	11

TAB. 2.1 – Principe de parcimonie - Nombre minimal d'événement évolutifs nécessaires pour expliquer chaque caractère pour les trois topologies possibles

2.5 Méthodes utilisant le maximum de vraisemblance

Les méthodes utilisant le maximum de vraisemblance tiennent également compte de la longueur des branches de l'arbre phylogénétique, en plus d'être basées sur les caractères.

2.5.1 Principe

Il s'agit, pour ces méthodes, de comparer un grand nombre d'arbre phylogénétiques possibles (topologie et longueurs des branches). L'arbre choisi sera celui qui maximise la probabilité d'observer les données utilisées, étant donné un modèle de substitution de l'ADN. Cette probabilité est appelée « vraisemblance » de l'arbre phylogénétique (équation 2.12).

$$L = P(D|T, \lambda) \tag{2.12}$$

avec

T un arbre possible

L la vraisemblance de l'arbre T

D les données observées

λ un modèle de substitution de l'ADN

On veut donc trouver l'arbre de topologie Θ, avec les longueurs de branches ν ayant la plus haute vraisemblance.

Pour ce faire, il faut construire tous les arbres possibles et comparer leur vraisemblance. Malheureusement, la détection, parmi tous les arbres possibles, de l'arbre de plus haute vraisemblance est un problème « NP-dur », c'est à dire qu'aucun algorithme connu ne peut résoudre ce problème en un temps polynomial.

En effet, le nombre de topologies possibles pour un arbre phylogénétique non raciné de t taxa est :

$$B(t) = \prod_{i=3}^{t}(2i-5) = \frac{(2t-5)!}{(t-3)!\,2^{t-3}} \tag{2.13}$$

Ce qu'il faudrait encore multiplier par toutes les longueurs de branches possibles pour obtenir le nombre total d'arbres possibles.

Même en ne tenant compte que des topologies, le nombre de possibilités augmente de manière exponentielle avec le nombre de taxa considérés (figure 2.11). Par conséquent, il est impossible de tester chacun des arbres possibles.

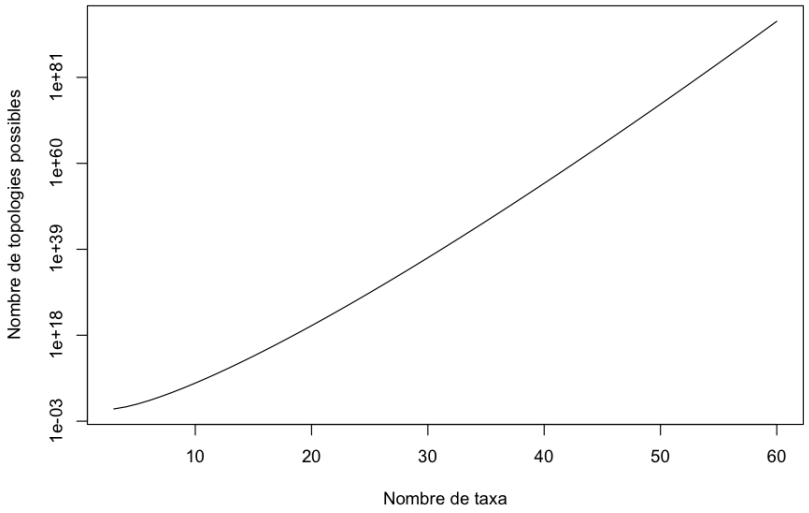


FIG. 2.11 – Illustration de la nature « NP-dure » de la recherche de l'arbre ayant la vraisembance maximale. L'axe vertical est logarithmique. Le nombre de topologies possibles augmente donc de manière exponentielle avec l'augmentation du nombre de taxa considéré.

On utilise donc des heuristiques, c'est-à-dire des algorithmes qui vont explorer une partie de l'espace des solutions en se concentrant principalement sur les parties « prometteuses ». Ces algorithmes permettent de trouver, en un temps polynomial, une bonne solution. Ils ne peuvent cependant pas garantir que la solution trouvée est la solution optimale. En effet, l'espace des solutions peut contenir de nombreux pics (régions de solutions de haute vraisemblance), séparés par de grandes vallées (régions de solutions de basse vraisemblance), et lorsque l'on s'arrête en haut d'un pic, mêmes si les solutions environnantes paraissent moins bonne, il est impossible de savoir avec certitude qu'aucun pic plus haut ne se trouve à quelques vallées de là.

Les méthodes de maximum de vraisemblances donnent de meilleurs résultats que les méthodes de distances et de parcimonie et présentent comme avantage d'avoir un cadre statistique consistant, permettant de comparer facilement les hypothèses. La contrepartie de cela étant bien sûr un temps de calcul bien plus élevé que ces deux méthodes.

A noter que l'utilisation des heuristiques ne se limite pas au critère de maximum de vraisemblance. Elles peuvent être également appliquées (et ont été appliquées dans la plupart des cas) aux méthodes utilisant la parcimonie.

2.5.2 Calcul de la vraisemblance

La vraisemblance L d'un arbre est définie comme la probabilité qu'a cet arbre de générer les données observées, étant donné un modèle de substitution de l 'ADN (équation 2.12).

On prend comme hypothèses au départ que (Felsenstein, 2004) :

- l'évolution des différents sites est indépendante
- l'évolution des différents lignages est indépendante

La première hypothèse nous permet de décomposer la vraisemblance de l'arbre en le produit des vraisemblances de l'arbre pour chaque site indépendant.

Si l'on a un ensemble de séquences alignées de longueur m, la vraisemblance peut donc être calculée comme suit :

$$L = \prod_{i=1}^{m} P(D^i|T) \tag{2.14}$$

où D^i est le i^e site des séquences alignées.

Il suffit donc de pouvoir calculer la vraisemblance site par site. Or, les modèles de substitutions de l'ADN (section 2.2) nous permettent de calculer $P_{ij}(t)$, la probabilité qu'un site soit dans l'état j après une branche de longueur t, si l'état de départ est i.

La vraisemblance se calcule à partir d'un arbre raciné. Cependant, l'endroit où cet arbre est raciné n'influence pas sur la valeur finale de sa vraisemblance.

Considérons un arbre, et les données observées pour un site (figure 2.12, exemple tiré de (Felsenstein, 2004)).

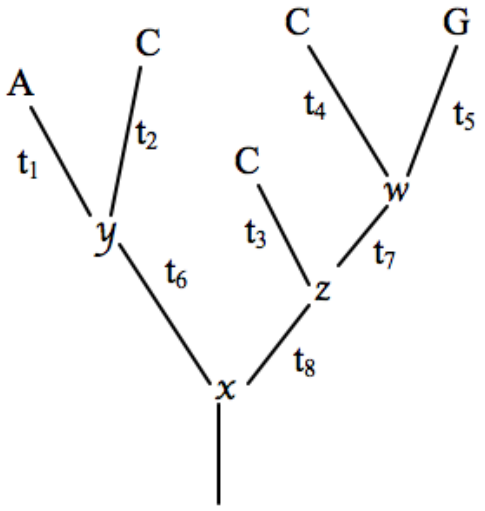


FIG. 2.12 – Arbre phylogénétique avec les longueurs des branches et les données observées pour un seul site, utilisé comme exemple pour calculer la vraisemblance

La vraisemblance de l'arbre pour ce site est la somme des probabilités des tous les scénarios menant à ces données, chaque noeud interne pouvant avoir été l'un des quatre nucléotides A,C,G ou T :

$$P(D^i|T) = \sum_x \sum_y \sum_z \sum_w P(A, C, C, C, G, x, y, z, w|T) \tag{2.15}$$

Chaque somme se faisant sur les quatre nucléotides possibles.

Le seconde hypothèse (indépendance des lignages) permet de décomposer la probabilité du terme de droite de l'équation 2.15 en un produit de plusieurs termes :

$$\begin{aligned} P(A, C, C, C, G, x, y, z, w|T) \quad &= P(x)\ P(y|x, t_6)\ P(A|y, t_1) \\ &\quad P(C|y, t_2)\ P(z|x, t_8)\ P(C|z, t_3) \\ &\quad P(w|z, t_7)\ P(C|w, t_4)\ P(G|w, t_5) \end{aligned} \tag{2.16}$$

Les différents termes de l'équation 2.16 sont calculés grâce au modèle de substitution de l'ADN choisi, à l'exception de $P(x)$ pour laquelle on peut considérer qu"il s'agit, pour chaque cas possible, de la probabilité à l'équilibre du nucléotide (toujours selon le modèle choisi).

Le nombre de termes à considérer pour calculer l'équation 2.16 augmente de manière exponentielle avec le nombre de taxa dans l'arbre phylogénétique. On utilise donc un algorithme de programmation dynamique pour pouvoir calculer la vraisemblance en un temps raisonnable.

La vraisemblance d'un sous-arbre k quelconque est calculée grâce à la vraisemblance des deux sous-arbres descendant directement de la racine du sous-arbre k. Donc, si l'on a un noeud k, ayant deux descendant l et m, les branches reliant ces deux noeuds à k étant de longueurs t_l et t_m respectivement, la probabilité des données observées pour un site i, étant donné le sous-arbre de racine k et k étant dans l'état s, $L_k^i(s)$ est calculée comme suit :

$$L_k^i(s) = \left(\sum_x P(x|s,t_l)L_l^i(x)\right)\left(\sum_y P(y|s,t_m)L_m^i(y)\right) \tag{2.17}$$

Et lorsque l'on arrive aux feuilles de l'arbre, les valeurs de L^i pour chaque feuille sont :

$$\big(L^i(A), L^i(T), L^i(C), L^i(G)\big) = \begin{cases} \text{si le site est dans l'état A} & (1,0,0,0) \\ \text{si le site est dans l'état T} & (0,1,0,0) \\ \text{si le site est dans l'état C} & (0,0,1,0) \\ \text{si le site est dans l'état G} & (0,0,0,1) \end{cases} \tag{2.18}$$

On a donc l'algorithme de récursion donné par l'algorithme 2.1.

Algorithm 2.1 Calcul de la vraisemblance

```
if node has no children then
    return likelihood equation 2.18
else
    left = ∑ likelihood(left child node) on the four possible states
    right = ∑ likelihood(right child node) on the four possible states
    return left x right
end if
```

2.5.3 Hill-Climbing

Le Hill-Climbing (HC) (Guindon and Gascuel, 2003; Mitchell and Holland, 1993) est la méthode heuristique la plus simple : on construit un arbre initial (avec une topologie, des longueurs de branches, et des paramètres du modèle de substitution ADN choisi), grâce à une méthode de distance par exemple, et on optimise ensuite la longueur de ses branches. On modifie alors la topologie de cet arbre, et on réoptimise les longueurs de branches. Si ce nouvel arbre a une vraisemblance supérieure à l'arbre initial, ce dernier est abandonné et remplacé par le nouveau. Et on recommence le processus : modification, optimisation, comparaison, conservation du meilleur arbre (algorithme 2.2), jusqu'à ce qu'aucune modification de la solution courante ne puisse améliorer celui-ci.

A chaque étape de l'algorithme, on optimise donc les longueurs de branches de l'arbre considéré (optimisation « intra-step »). De ce fait, l'algorithme atteint un optimum en peu d'étapes, et est donc très rapide. Par contre, beaucoup de calculs sont fait pour optimiser des arbres phylogénétiques qui seront rejetés tout de suite après car moins bons que l'arbre précédent.

Un autre désavantage de cet algorithme est qu'en n'acceptant uniquement les changements qui améliorent l'arbre phylogénétique, on n'explore l'espace des solution que très localement. Par conséquent, on reste bloqué dans un optimum local, qui peut être beaucoup moins bon que l'optimum global.

Algorithm 2.2 Algorithme Hill-Climbing

```
bestTree = Generate solution
Optimise branch length of the bestTree
while a rearrangement can be found that improves current best tree do
  newTree = modification of the topology of bestTree
  Optimise branch length of the newTree
  if likelihood(newTree) > likelihood(bestTree) then
    bestTree = newTree
  end if
end while
```

2.5.4 Recuit Simulé

L'algorithme de Recuit Simulé (SA) (Kirkpatrick *et al.*, 1983; Brooks and Morgan, 1995; Salter and Pearl, 2001) tombe moins facilement dans des optima locaux que le Hill-climbing en permettant la sélection régulière d'une solution moins bonne que la solution courante.

Le nom de recuit simulé (en anglais *simulated annealing*) vient de la thermodynamique et fait référence à la cristallisation de certains liquides et métaux.

Lorsque l'on refroidit une substance en vue d'obtenir un cristal, cette substance est au départ dans un état non-ordonné, avec une haute énergie libre. Au fur et à mesure que la température diminue, les molécules s'ordonnent et s'alignent. Lorsqu'elles sont complètement alignées, et forment donc un cristal, la substance a atteint un état d'énergie minimale. Cependant, si la température décroît trop rapidement, certaines molécules peuvent être figées dans des positions non-optimales, et le cristal présentera des défauts (ou la substance ne cristallisera pas du tout). Il est donc essentiel de diminuer la température très lentement et par paliers (ce processus est alors appelé « annealing »), afin de laisser le temps aux molécules de se mettre en place.

L'algorithme de SA est donc basé sur un principe similaire. On veut atteindre une solution optimale, c'est-à-dire de vraisemblance maximale (correspondant à une énergie mimimale). Pour ce faire, on modifie la solution en permettant éventuellement des changements qui n'améliorent pas immédiatement la solution. C'est-à-dire qu'à la différence du Hill-Climbing, où l'on n'accepte que les modifications qui améliorent la solution, les solutions de vraisemblance inérieure à la solution courante seront acceptées avec une certaine probabilité. Et cette probabilité est proportionnelle à une variable nommé « température », qui est haute au début de l'algorithme et descend ensuite par paliers.

Ainsi, au départ, la probabilité d'accepter une « moins bonne »solution sera très grande, et au fur et à mesure des itérations de l'algorithme, cette probabilité diminuera. De ce fait, il sera possible de traverser des « vallées » de l'espace des solutions, alors que le Hill-Climbing ne peut que grimper une colline (la première qu'il rencontre).

Par ailleurs, l'algorithme de recuit simulé ne fait plus une optimisation « intra-step » (optimisation des longueurs de branches à chaque itération, avant la comparaison avec la meilleure solution courante) mais bien une optimisation inter-step. A chaque itération, il y a modification à la fois de la topologie et/ou de la longueur des branches, et on n'optimise plus la longueur des branches avant de comparer le nouvel arbre à l'arbre courant. On évite ainsi de longs calculs sur des arbres qui seront rejetés.

L'algorithme 2.3 décrit de manière simplifiée le recuit simulé.

Algorithm 2.3 Algorithme Recuit Simulé

```
bestTree = Generate solution
while a rearrangement can be found that improves current best tree do
  newTree = modification (topology OR branch length) of bestTree
  if likelihood(newTree) > likelihood(bestTree) then
    bestTree = newTree
  else
    compute P(keep newTree)
    if random number in {0-1} < P(keep newTree) then
      bestTree = newTree
    end if
  end if
end while
```

2.5.5 Algorithme Génétique

Un Algorithme Génétique (GA) est un algorithme qui s'inspire des mécanismes de l'évolution biologique (De Jong, 1988) : mutations, sélection des individus les plus aptes, reproduction avec recombinaison.

L'algorithme 2.4 donne les grandes étapes d'un algorithme génétique.

On commence par générer plusieurs arbres phylogénétiques (dont on définit la topologie, la longueur des branches ainsi que les paramètres du modèle de substitution ADN choisi), pour former une *population de solutions* (la génération 0). Chaque *individu* de cette population est alors évalué grâce à une *fonction de fitness*. Dans le cas présent, la fitness est bien sûr proportionnelle à la vraisemblance de l'arbre (on utilise en général le logarithme de la vraisemblance).

On va alors créer une nouvelle population de solutions à partir de la génération 0 : les individus ayant les meilleurs fitness sont sélectionnés pour être les « parents » de la génération suivante. Ils sont alors recombinés entre eux (typiquement deux par deux) pour former de nouveaux individus. Pour terminer, ces individus subissent des mutations au hasard.

La nouvelle population est alors à sont tour évaluée, et l'on continue itérativement à créer de nouvelles générations. Comme les individus de la génération n sont formés à partir des « meilleurs » individus de la génération $n-1$, la fitness moyenne des individus augmente de génération en génération jusqu'à atteindre un optimum, qu'on espère global.

Il existe de nombreuses variantes des GAs, et ce, pour chaque étape de l'algorithme. La sélection peut se faire par exemple de manière stricte (on prend les x meilleurs individus) ou stochastique (la probabilité qu'à un individu de se reproduire est proportionnelle à sa fitness), la recombinaison peut se faire de diverses manière, les conditions d'arrêts peuvent également varier (arrêt après un certain temps, un certain nombre d'itérations, un certain nombre d'itérations sans amélioration de la fitness...).

Algorithm 2.4 Algorithme Génétique

```
Initialize population
Evaluate initial population
while stopping conditions not met do
  Select best individuals
  Perform crossover on selected individuals
  Mutate new individuals
  Evaluate new population
end while
```

2.6 Méthodes Bayesiennes

Les méthodes Baysiennes sont dérivées des méthodes de maximum de vraisemblance.

Etant donné un arbre possible H et les données observées D, on définit (grâce au théorème de Bayes) la probabilité postérieure :

$$P(H|D) = \frac{P(H,D)}{P(D)} = \frac{P(H)P(D|H)}{\sum_H P(H)P(D|H)} \tag{2.19}$$

$P(D|H)$ étant la vraisemblance de l'arbre H (voir section 2.5 equation 2.12).

Le dénominateur de l'équation 2.19 est impossible à calculer, étant donné qu'il faudrait faire une somme sur toutes les topologies et toutes les longueurs de branches possibles. Il faut donc utiliser une méthode permettant de faire une approximation de la probabilité postérieur : une Chaîne de Markov Monte Carlo (MCMC).

La méthode MCMC la plus utilisée est l'algorithme de Metropolis-Hastings (MH) (Metropolis *et al.*, 1953; Hastings, 1970; Green, 1995; Huelsenbeck *et al.*, 2001) avec lequel on va échantillonner l'espace des solutions.

L'algorithme de MH construit une chaîne de Markov constituée d'arbres phylogénétiques de l'espace des solutions. L'arbre courant est noté Ψ (initialisation au hasard). A chaque itération de l'algorithme, on propose un nouvel arbre $\Psi\prime$, et ce nouvel arbre est accepté (donc $\Psi = \Psi\prime$) avec une probabilité R donnée par :

$$\begin{aligned} R &= min\left(1, \frac{P(\Psi\prime|D)}{P(\Psi|D)} \times \frac{P(\Psi|\Psi\prime)}{P(\Psi\prime|\Psi)}\right) && (2.20)\\ &= min\left(1, \frac{P(D|\Psi\prime)P(\Psi\prime)/P(D)}{P(D|\Psi)P(\Psi)/P(D)} \times \frac{P(\Psi|\Psi\prime)}{P(\Psi\prime|\Psi)}\right) && (2.21)\\ &= min\left(1, \frac{P(D|\Psi\prime)}{P(D|\Psi)} \times \frac{P(\Psi\prime)}{P(\Psi)} \times \frac{P(\Psi|\Psi\prime)}{P(\Psi\prime|\Psi)}\right) && (2.22) \end{aligned}$$

Cela est répété plusieurs milliers de fois, et la séquence des arbres visités forme une chaîne de Markov. On échantillonne alors cette chaîne, et la proportion de tirage d'un arbre est une approximation acceptable de la probabilité postérieure de cet arbre (Tierney, 1994).

Tout comme pour la vraisemblance, la distribution des probabilités postérieures contient de nombreux pics et vallées. Les méthodes Bayesiennes peuvent donc également être bloquées dans un optimum local.

2.7 Autres méthodes

D'autres heuristiques ont été utilisées pour l'inférence phylogénétique (liste non exhaustive) :

Optimisation par colonie de fourmis (Catanzaro *et al.*, 2007) Cette heuristique consiste en la construction de solution partielle de manière itérative par des « fourmis virtuelles ». Les fourmis déposent des « phéromones » sur les branches des solutions (plus une solution est bonne, plus elle reçoit de phéromones). L'ajout d'une branche à une solution partielle se fait en fonction de la concentration de phéromones de cette branches. Les branches appartenant à de bonnes solutions ont donc plus de chances d'être utilisées par les fourmis suivantes. A terme, les solutions construites par les fournis convergent vers un optimum.

Quartet puzzling (Strimmer and von Haeseler, 1996; Schmidt and von Haeseler, 2007) Pour chaque quadruplet de taxa, un arbre est construit. On reconstruit ensuite l'arbre entier en respectant au maximum les arbres des quadruplets.

Hitch-Hiking (Charleston, 2001), **Structural Expectation Maximization** (Friedman *et al.*, 2002), **Tabu search** (Lin, 2008)...

Chapitre 3

Présentation du programme MetaPIGA

MetaPIGA (Lemmon and Milinkovitch, 2002) est un programme qui, à partir de séquences ADN alignées (au format NeXus), va construire un arbre phylogénétique pour ces séquences. La première version utilisait une variante des algorithmes génétiques : l'algorithme génétique métapopulationnel (voir section 3.1.4), qui a donné son nom au programme (***P**hylogenetique **I**nference using the **META**population **G**enetic **A**lgorithm*).

La seconde version, encore non publiée, permet de choisir entre plusieurs heuristiques pour inférer l'arbre phylogénétique. Lors de la rédaction de ce travail, certains outils restaient encore à programmer.

3.1 Heuristiques disponibles

3.1.1 Hill-Climbing

L'algorithme de HC implémenté dans MetaPIGA est une variante de l'algorithme décrit section 2.5.3.

En effet, il optimise les paramètres de manière inter-step, et non plus intra-step. C'est-à-dire qu'à chaque itération, la topologie OU la longueur d'une branche OU un paramètres du modèle de substitution ADN va être modifié, alors que dans le HC classique, seule la topologie est modifié, et les longueurs de branches et paramètres sont optimisés complètement après chaque modification de l'arbre.

De cette manière, l'algorithme fera plus d'itérations avant que la vraisemblance ne se stabilise, mais on évite de nombreux calculs inutiles et coûteux en terme de temps et de mémoire sur des arbres qui seront rejetés directement.

3.1.2 Recuit Simulé

MetaPIGA implémente un algorithme de SA s'inspirant fortement de (Salter and Pearl, 2001), décrit section 2.5.4.

Options et paramètres du SA

La diminution de la température au cours du temps (« cooling schedule ») peut se faire de nombreuses manières.

Soient T_i la température après i décréments, T_0 la température de départ et T_Γ la température minimale (T_0 et T_Γ ne sont pas toujours définies), Γ étant alors le nombre maximum de décréments

de la température avant de la réinitialiser à T_0 (voir p.21), les schéma suivants peuvent être appliqués à la diminution de la température au cours du temps :

• **Lundy** (Lundy, 1985) $T_{i+1} = \frac{\Delta L}{1+i\beta}$ avec

ΔL une limite supérieure pour la diminution de la vraisemblance par itération

$\beta = \frac{c}{(1-\alpha)n+\alpha\frac{-\ln NJT}{m}} < 1$

n le nombre de séquences

m le nombre de sites

c et α paramètres à définir, entre 0 et 1

$\ln NJT$ logarithme népérien de la vraisemblance de l'arbre phylogénétique construit par NJ

• **Ratio-Percent** $T_{i+1} = \delta T_i$ avec $0 < \delta < 1$

• **Fast Cauchy** $T_i = \frac{T_0}{i}$

• **Boltzmann** $T_i = \frac{T_0}{\ln i}$

• **Diminution géométrique** $T_i = T_0\alpha^i$ avec $\alpha < 1$

• **Diminution linéaire** $T_i = T_0 - i\frac{T_0-T_\Gamma}{\Gamma}$

• **Diminution triangulaire** $T_i = T_0\left(\frac{T_0}{T_\Gamma}\right)^{\frac{i}{\Gamma}}$

• **Diminution polynomiale** $T_i = \frac{(T_0-T_\Gamma)(\Gamma+1)}{\Gamma(i+1)} + T_0 - \frac{(T_0-T_\Gamma)(\Gamma+1)}{\Gamma}$

• **Diminution exponentielle transcendentale** $T_i = T_\Gamma + \frac{T_0-T_\Gamma}{1+e^{3(i-\Gamma/2)}}$

• **Diminution logarithmique transcendentale** $T_i = T_0 e^{-\left(\frac{i}{\Gamma}\right)^2 \ln\frac{T_0}{T_\Gamma}}$

• **Diminution périodique transcendentale** $T_i = \frac{T_0-T_\Gamma}{2}\left(1+\cos i\frac{\pi}{\Gamma}\right) + T_\Gamma$

• **Diminution périodique transcendentale ralentie** $T_i = \frac{T_0-T_\Gamma}{4}\left(2+\cos 8i\frac{\pi}{\Gamma}\right)e^{-\frac{i}{2\Gamma}}$

• **Diminution selon une tangente hyperbolique** $T_i = \frac{T_0-T_\Gamma}{2}\left(1-\tanh\left(\frac{10i}{\Gamma}-5\right)\right) + T_\Gamma$

• **Diminution selon un cosinus hyperbolique** $T_i = \frac{T_0-T_\Gamma}{\cosh\frac{10i}{5}+T_\Gamma}$

Selon le cooling schedule choisi, différents paramètres sont à définir : c et α pour *Lundy* ; *initial acceptance* et *final acceptance*, qui sont les probabilité initiale et finale d'accepter un arbre de vraisemblance inférieure, nécessaires pour calculer T_0 et T_Γ avec les diminutions linéaire, triangulaire, polynomiale, exponentielle, logarithmique, périodique, périodique lissée et selon une tangente et un cosinus hyperbolique.

Pour tous les cooling schedule, à l'exception de *Lundy*, T_0 et T_Γ (si nécessaire) sont calculés de la manière suivante :

$$T_0 = \left|\frac{-\Delta L}{\ln A_0}\right| \tag{3.1}$$

$$T_\Gamma = \left|\frac{-\Delta L}{\ln A_\Gamma}\right| \tag{3.2}$$

ΔL étant la diminution de vraisemblance qui sera accepté avec une probabilité de A_0 ; A_0, la probabilité initiale (*initial acceptance*) d'accepter un arbre dont la vraisemblance est moins haute de ΔL que celle de l'arbre courant et A_Γ la probabilité finale (*final acceptance*) d'accepter un arbre dont la vraisemblance est moins haute de ΔL que celle de l'arbre courant.

La valeur de ΔL peut être calculée selon deux méthodes :

- Période « burn-in » : Chaque opérateur sélectionné (voir section 3.3) est appliqué 20 fois à l'arbre de départ (voir section 3.4) et la plus grande différence de vraisemblance observée pendant cette période est considérée comme le ΔL
- Pourcentage du NJ : ΔL vaut un certain pourcentage p (choisi par l'utilisateur) de la vraisemblance de l'arbre construit grâce à l'algorithme de NJ

Le nombre d'itérations de l'algorithme entre chaque diminution de la température est un paramètre réglable. Dans l'algorithme décrit par (Salter and Pearl, 2001), la température diminue à chaque itération. MetaPIGA permet de choisir entre :

- Diminution de la température toutes les x étapes
- Diminution de la température après x améliorations ou y diminutions de la vraisemblance

x et y étant choisis par l'utilisateur.

Cela permet d'explorer plusieurs possibilités à chaque température, augmentant les capacités d'exploration de l'espace des solutions, notamment dans le cas de cooling schedule faisant changer drastiquement la température en quelques décréments.

Le paramètre de remise de la température à T_0 est le dernier paramètre du SA de MetaPIGA. On a le choix entre :

- Aucune remise à T_0
- Remise à T_0 après x décréments de la température
- Remise à T_0 lorsque le température atteint $x\%$ de la température de départ T_0

x étant choisi par l'utilisateur.

Remonter la température permet d'éventuellement sortir d'un optimum local. En effet, lorsque la température est très basse, la probabilité d'accepter une solution de moindre vraisemblance est tellement petite que l'on est presque dans un HC. Remettre la température à T_0 va donc permettre de repartir explorer l'espace des solutions.

3.1.3 Algorithme génétique

Au moment de la rédaction de ce travail, l'algorithme génétique implémenté dans MetaPIGA ne recombine pas les arbres phylogénétique sélectionnés pour former la génération suivante. Ce n'est donc pas un « vrai »algorithme génétique pour l'instant.

Toutes les autres étapes de l'algorithme sont telles qu'expliquées dans la section 2.5.5

Options et paramètres du GA

Le nombre d'individus dans la population est bien sûr un paramètre important du GA. Le minimum est de deux individus et on est limité à 100 bien que, potentiellement, le nombre d'individus utilisés dans un algorithme génétique est infini.

La méthode de sélection des individus qui formeront la génération suivante peut être une :

- **Sélection par rang** Les individus sont classés par ordre décroissant de vraisemblance, et à chaque individu est assignée une probabilité proportionnelle à son rang de laisser une descendance. L'individu au i^e rang aura une probabilité $P_i = \frac{2}{n(n+1)}(n-i+1)$ de laisser une descendance. Les individus qui ne laissent pas de descendance sont remplacés par une copie du meilleur individu
- **Sélection par tournoi** Deux individus sont tirés au hasard dans la population G_n, et une descendance est générée par l'individu de plus haute vraisemblance. Les deux arbres sont ensuite replacés dans la population G_n, et on recommence le processus jusqu'à avoir une population G_{n+1} (qui doit encore subir des mutations) de même taille que G_n
- **Sélection par remplacement** Deux individus sont tirés au hasard dans la population G_n. L'individu de plus basse vraisemblance est remplacé par celui de plus haute vraisemblance dans la population G_n. Le processus est répété sn fois (s est la force de la sélection, choisie par l'utilisateur, et n est le nombre d'individus dans la population). Une fois ces sn remplacements d'un « plus faible » par un « plus fort » dans la population G_n, on génère la population G_{n+1} en copiant simplement la génération G_n (la population G_{n+1} subira ensuite des mutations)
- **Sélection des améliorations** Seuls les arbres ayant une vraisemblance plus haute que le meilleur arbre de la génération précédente sont conservés, les autres sont remplacés par la meilleure solution courante

• **Sélection du meilleur** Seul le meilleur individu de la population est sélectionné, les autres individus sont remplacés par une copie de celui-ci

L'application des opérateurs de mutation (voir section 3.3) peut se faire de deux manières :

• **Appliqué à la population** Lors de la mutation des individus le même opérateur est appliqué à tous les individus de la population

• **Appliqué à l'individu** un opérateur est appliqué séparément à chaque individu

3.1.4 Algorithme génétique métapopulationnel

MetaPIGA implémente un algorithme génétique métapopulationnel, dont le principe est le suivant :

Plusieurs populations d'arbres phylogénétiques sont générées, et à chacune de ces populations va être appliqué un algorithme génétique (voir sections 2.5.5 et 3.1.3).

L'intérêt d'avoir plusieurs populations évoluant en parallèle, est qu'on peut faire interagir ces populations, en appliquant le « Consensus Pruning (CP) » : toutes les x générations (x étant défini à l'avance ou un nombre tiré au hasard), on prend le « meilleur » arbre de chaque population, et on compare ces arbres, afin de voir s'ils ont des branches en commun. Si une branche se retrouve dans toutes les populations, elle est fixée, et les mutations appliquées à chaque génération d'arbres ne peuvent la modifier (voir Figure 3.1).

Le CP permet d'appliquer une forte sélection aux populations, sans risquer de se retrouver bloqué rapidement dans un optimum local. En effet, la probabilité que plusieurs populations parviennent au même optimum local est très faible (et diminue bien sûr avec le nombre de populations).

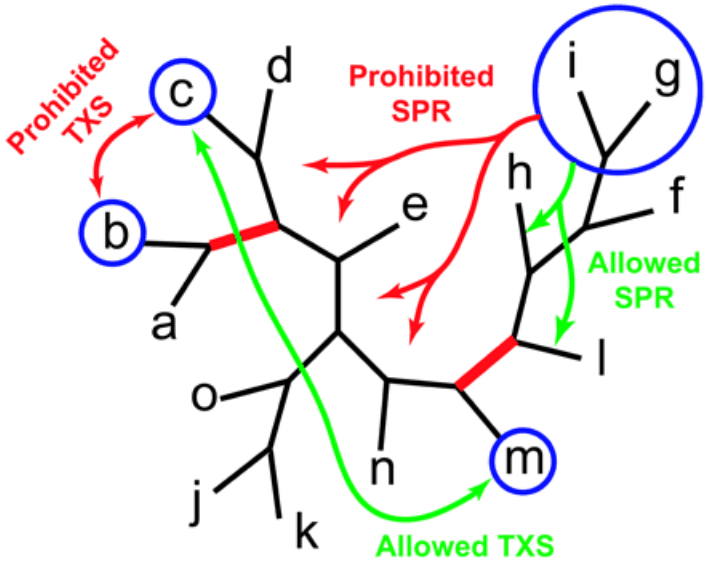


FIG. 3.1 – Consensus Pruning. (Lemmon and Milinkovitch, 2002) Le meilleur arbre de chaque population isole les taxa a et b du reste des taxa, ainsi que les taxa f, g, h, i et l. Les branches isolant ces groupes de taxa (en rouge) ne peuvent donc plus être modifiées. Les flèches rouges représentent des mutations non autorisées, les flèches vertes des mutations autorisées

Options et paramètres du CP

Les **méthode de sélection des individus**, et le **nombre d'individus par population** correspondent aux paramètres du GA

L'application des opérateurs de mutations possède une option supplémentaire par rapport au GA :

- **Appliqué à toutes les populations** Le même opérateur est appliqué à toutes les populations, donc à tous les individus de toutes les populations
- **Appliqué à la population** Un opérateur est appliqué séparément à chaque population (et donc à tous les individus de cette population)
- **Appliqué à l'individu** Un opérateur est appliqué séparément à chaque individu

Enfin, trois paramètres liés au consensus sont à définir :

Le type de consensus peut être

- **Strict** Les branches communes au meilleur arbre de chaque population (100% de consensus) ne peuvent plus être mutées. Il n'y a aucune contrainte sur les mutations des autres branches.
- **Stochastique** A chaque branche commune à au moins deux des meilleurs arbres de chaque population est assignée une valeur de consensus (VC) correspondant à la proportion des meilleurs arbres possédant cette branche. La probabilité de mutation de ces branches est de $1 - VC$.

La sélection des opérateurs peut être définie comme

- **Aveugle** Lors de l'étape de mutation, un opérateur (voir section 3.3) est sélectionné au hasard. Si l'application de cet opérateur casse une branche consensus, la mutation ne se fait pas et l'arbre reste inchangé. Au fur et à mesure de l'avancement de l'algorithme, de plus en plus de branches ne pourront être cassées, et il arrivera donc de plus en plus souvent qu'aucune mutation ne se fasse.
- **Supervisée** Lors de l'étape de mutation un opérateur est choisi au hasard *parmi un ensemble d'opérateurs ne cassant aucune branche consensus*. Si cet ensemble est vide, l'arbre demeure inchangé. L'étape de mutation durera donc plus longtemps mais les chances qu'aucune mutation ne se fasse sont très réduites.

Et un paramètre de tolérance permet de passer outre les consensus et donc éventuellement de sortir d'un optimum local dans lequel les contraintes de consensus bloquaient l'algorithme. Il est défini comme une probabilité de faire une mutation sans tenir compte du consensus (qu'il soit de type strict ou stochastique).

3.2 Modèles de substitution de l'ADN

MetaPIGA permet de choisir entre les cinq modèles de mutations décrits section 2.2 : Jukes-Cantor, Kimura-2-paramètres, Hasegawa-Kishino-Yano (1985), Tamura-Nei (1993) et General Time Reversible, auxquels peuvent être ajoutées une distribution gamma et/ou une proportion d'invariants.

Ces modèles servent non seulement à calculer la vraisemblance des arbres, mais peuvent également être utilisés pour construire les arbres de départ (voir section 3.4).

Lorsqu'une distribution gamma est définie, celle-ci est approximée par une discrétisation de cette distribution en plusieurs catégories. Ceci dû au fait qu'il faudrait, pour calculer la vraisemblance sans discrétisation, intégrer la fonction de vraisemblance sur l'ensemble des valeurs de la distribution, ce qui requiert un énorme temps de calcul et n'est pas faisable pour des phylogénies de plus de quelques taxa (Yang, 1994).

Le nombre de catégories de taux utilisées pour approximer la distribution gamma est de quatre pour le modèle servant à la construction des arbres initiaux, et peut être choisi par l'utilisateur pour le modèle servant à l'évaluation des arbres.

3.3 Opérateurs

Dix opérateurs différents sont utilisés par MetaPIGA pour modifier les arbres phylogénétiques. Deux modifient la longueurs des branches, cinq changent la topologie et trois modifient les paramètres du modèle de substitution de l'ADN.

L'utilisateur peut choisir quels sont les opérateurs qui seront utilisés, et comment ils seront sélectionnés parmi ces trois options :

Au hasard : à chaque modification d'un arbre, un opérateur est choisi au hasard.

Ordonné : les opérateurs sont sélectionnés les uns après les autres.

Par fréquence à chaque opérateur est assigné une fréquence (choisie au départ par l'utilisateur). A chaque modification d'un arbre, les opérateurs ont une probabilité d'être sélectionnés correspondant à leur fréquence. Lorsque cette option est seléctionnée, la fréquence de chacun des opérateurs peut être définie comme **dynamique**. Dans ce cas, toutes les x étapes de l'algorithme (x donné par l'utilisateur), cette fréquence sera adaptée en fonction de l'efficacité de l'opérateur. C'est-à-dire que plus un opérateur a fait augmenter la vraisemblance des arbres auxquels il est appliqué, plus sa fréquence sera augmentée, et vice versa. Une borne inférieure (règlable) sur la fréquence permet d'éviter que certains opérateurs ne tombent à une fréquence de zéro, et ne soient donc plus jamais sélectionnés.

3.3.1 Opérateurs modifiant les longueurs des branches

Mutation de la longueur des branches (BLM) (*« Branch Length Mutation »*) : modifie au hasard n'importe quel branche, en multipliant sa longueur par un facteur tiré d'une distribution exponentielle

$$P(x) = \lambda e^{-\lambda x} \text{ avec } x \geq 0 \tag{3.3}$$

de moyenne égale à 1 (donc $\lambda = 1$) et décalée de 0,5 vers la droite (la branche modifiée aura donc une longueur d'au moins la moitié de sa longueur d'origine)

Mutation de la longueur des branches internes (BLMINT) (*« Branch Length Mutation on INTernal branch only*) : modifie au hasard l'une des branches *internes* de l'arbre de la même manière que l'opérateur BLM

3.3.2 Opérateurs affectant la topologie des arbres

Un exemple de chacun de ces opératurs est donnée figure 3.2

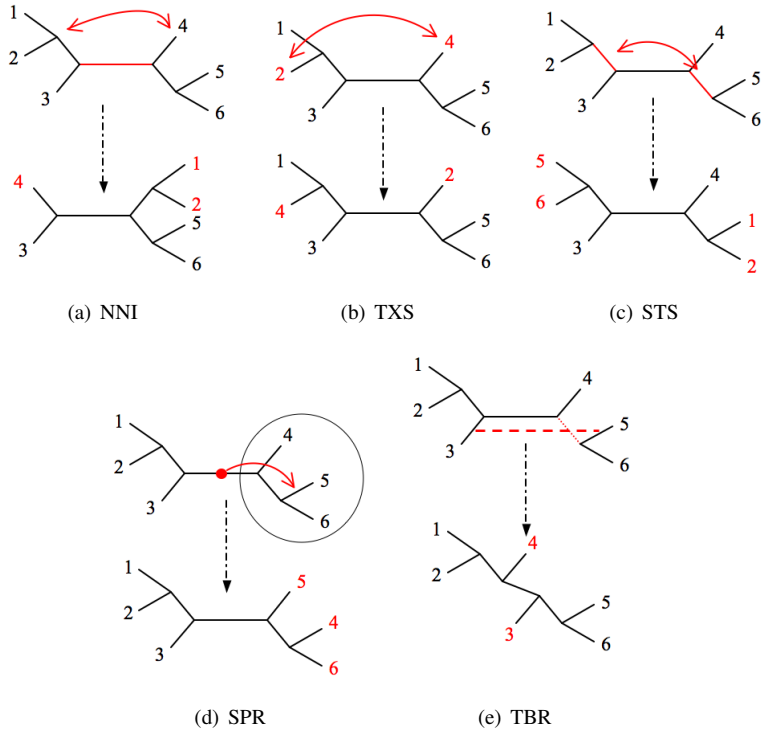
Echange des plus proches voisins (NNI) (*« Nearest Neighbour Interchange »*) : permutation de deux sous-arbres autour d'une branche interne.

Echange de taxa (TXS) (*« Taxa Swap »*) : permutation de plusieurs taxa. L'utilisateur peut choisir de permuter entre deux et tous les taxa, ou de permuter un nombre au hasard de taxa. Dans le cas où l'on ne permute que deux taxa, le programme évite la permutation entre taxa frères (qui ne change pas la topologie). Par contre, lorsque l'on permute plus de deux taxa, aucune vérification n'est faite, par conséquent, certaines permutations sont inutiles (on ne permute donc pas x taxa mais *jusqu'à* x taxa pour $x > 2$).

Echange de sous-arbres (STS) (*« Sub-tree Swap »*) : permutation de plusieurs sous-arbres. L'utilisateur peut choisir de permuter deux sous-arbres ou d'en permuter un nombre au hasard. Comme pour le TXS, le programme évite les permutation entre sous-arbres frères uniquement lorsqu'il n'y permutation qu'entre deux sous-arbres.

Détachement et rattachement d'un sous-arbre (SPR) (*« Sub-tree Pruning and Regrafting »*) : un sous-arbre est détaché de l'arbre et rattaché sur une autre branche.

Division et reconnection d'un arbre (TBR) (*« Tree Bissection - Reconnection »*) : l'arbre est coupé en deux et ses deux parties sont reconnectées différemment



(a) NNI (b) TXS (c) STS

(d) SPR (e) TBR

FIG. 3.2 – Opérateur affectant la topologie d'un arbre phylogénétique. (a)NNI : on permute deux sous-arbres autour de la branche sélectionnée (en rouge). (b) TXS de deux taxa : deux taxa sont sélectionnés (en rouge) et permutés. (c) STS de deux sous-arbres : deux sous-arbres sont sélectionnés (partant des branches en rouge) et permutés. (d) SPR : un sous-arbre est sélectionné (entouré) et détaché de l'arbre. Il est ensuite rattaché à partir d'une autre branche (flèche rouge). (e) TBR : l'arbre est coupé en deux (suppression de la branche en fins pointillés rouges) et les deux partie sont reconnectées (larges pointillés rouges).

3.3.3 Opérateurs modifiant les paramètres du modèle de substitution de l'ADN

Mutation des paramètres de taux de substitution (RPM) (*« Rate Parameter Mutation »*) : selon le choix de l'utilisateur, modifie au hasard soit un, soit tous les taux de substitution (non utilisé si le modèle de substitution est JC)

Mutation du paramètre de la distribution gamma (GDM) (*« Gamma Distribution Mutation »*) : modifie au hasard le paramètre α (paramètre de forme) de la distribution gamma (utilisable uniquement si une distribution gamma a été définie par l'utilisateur)

Mutation de la proportion d'invariants (PIM) (*« Proportion of Invariant sites Mutation »*) : modifie au hasard la proportion de sites invariants (utilisable uniquement si une proportion initiale a été définie par l'utilisateur)

3.4 Génération des arbres initiaux

Les arbres utilisés comme point de départ des algorithmes (un seul arbre dans le cas du HC et du SA, plusieurs pour le GA et le CP) peuvent être de quatre sortes :

Arbre au hasard : création d'une topologie au hasard. Les longueurs des branches sont tirées au hasard dans une distribution exponentielle (équation 3.3) de paramètre λ égal à 1 et décalée de 0.01 vers la droite pour éviter d'avoir des branches de longueur nulle.

Arbre donné par l'utilisateur : l'utilisateur peut fournir un fichier contenant un (ou plusieurs) arbre(s) phylogénétique(s) au format NeXus. Si l'heuristique choisie est le HC ou le SA,

seul le premier arbre sera utilisé, pour le GA, les i premiers arbres (i étant le nombre d'individus), et s'il s'agit du CP, les p premiers arbres seront utilisé (p étant le nombre de populations choisi).

Neighbour-Joining : construction d'un arbre grâce à l'algorithme de NJ (voir section 2.3.2).

Loose Neighbour-Joining (LNJ) : variante du NJ permettant d'introduire du hasard dans la construction de l'arbre. Soit un arbre de n taxa, on définit un pourcentage P. L'algorithme de NJ classique joint les deux noeuds les plus proches (d'après la matrice de distances corrigée, voir section 2.3.2). Le LNJ va joindre hasard deux noeud ayant l'une des $P \times \frac{n \times n}{2}$ plus petites distances. Les longueurs de branches sont calculées de la même manière qu'avec le NJ classique.

Dans le cas du NJ et du LNJ, une matrice de distance est calculée. L'utilisateur peut donc choisir quel modèle de mutation de l'ADN sera utilisé pour construire celle-ci, et décider de l'utilisation ou non d'une distribution gamma et d'une proportion d'invariants. La distribution gamma est approximée avec quatre catégories de taux de substitution. Quant à la proportion d'invariants, après l'avoir définie, il faut également choisir la composition en bases nucléiques de ces sites invariants. Les trois choix possibles (Waddell and Steel, 1997) sont :

Fréquences égales : les quatre bases ont la même fréquence dans les sites invariants.

Fréquences estimées : les sites invariant ont la même composition en bases que les séquences prises dans leur ensemble.

Fréquences constantes : la composition en bases des sites invariants est estimée à partir des sites qui semblent n'avoir pas varié (donc qui gardent la même base dans toutes les séquences considérées).

3.5 Divers paramètres et options

3.5.1 Conditions d'arrêt des algorithmes

Trois conditions d'arrêt de l'algorithmes sont implémentées dans MetaPIGA :

Arrêt après x étapes L'algorithme s'arrête après un nombre x d'étapes défini par l'utilisateur

Arrêt après un temps t L'algorithme s'arrête après un temps t défini par l'utilisateur

Arrêt automatique L'algorithme s'arrête lorsqu'il n'y a pas eu d'amélioration de la vraisemblance après un nombre d'étapes défini par l'utilisateur

3.5.2 Options relatives aux taxa

Il est possible de définir un ou plusieurs taxa comme étant « l'ougroup », qui servira à enraciner l'arbre. Les taxa appartenant à l'outgroup formeront donc un sous-arbre, et les taxa dans « l'ingroup » un autre sous-arbre. Ces deux sous-arbres seront optimisé séparément par MetaPIGA.

Certains taxa peuvent également être exclus par l'utilisateur. Ils ne seront donc pas pris en compte par le programme.

3.5.3 Options relatives aux séquences ADN

Les « gaps », c'est-à-dire des sites pour lequels il n'y a aucune information sur la composition nucléotidique, peuvent être traités de deux manières : soit les sites possédant un gap dans au moins l'une des séquences sont éliminés, soit les gaps sont considérés comme des nucléotides N, donc A ou T ou C ou G.

Les séquences peuvent par ailleurs être séparées en plusieurs sets. Il est ensuite possible de ne pas tenir compte de certains sets pour inférer l'arbre, ou de traiter chaque set séparément, c'est-à-dire de leur assigner à chacun une matrice de taux de substitution (selon le modèle choisi), et

seconde base du codon

première base du codon	U		C		A		G	
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys
	UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop
	UUG	Leu	UCG	Ser	UAG	Stop	UGG	Trp
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
	AUG	Ile	ACG	Thr	AAG	Lys	AGG	Arg
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly

FIG. 3.3 – Code génétique universel (source : http ://www.didier-pol.net/).

éventuellement une distribution gamma et une proportion d'invariants. La vraisemblance sera donc calculée séparément pour chaque set, et on multipliera ces vraisemblances pour obtenir la vraisemblance de l'arbre entier.

Cela peut par exemple permettre de différencier les positions dans les codons (trio de nucléotides successifs déterminant un acide aminé). On sait en effet que les mutations en certains endroits de nombreux codons (souvent en troisième position) sont moins soumises à la pression de sélection, car plusieurs acides aminés sont définis par plusieurs codons différant uniquement par une base (figure 3.3). De ce fait, les taux de substitution peuvent être par exemple plus élevés pour la troisième position des codons que pour les deux premières.

On peut également ainsi différencier introns et exons des gènes, et/ou gènes et non-gènes.

Les calculs prendront plus de temps puisqu'il faut calculer la vraisemblance sur chacun des sets de caractères, mais les paramètres seront plus proches de la réalité, et on peut donc s'attendre à trouver de meilleurs arbres.

3.5.4 Réplicats

MetaPIGA permet répéter l'heuristique choisie plusieurs fois (nombre défini par l'utilisateur), et construit alors un arbre consensus avec les résultats obtenus à chaque application de l'heuristique. Les valeurs consensus (valeurs de support) de chacune des branches de l'arbre consensus, correspondant à la proportion d'arbres construits présentant cette branche, pourraient être l'équivalent des probabilités postérieures obtenues par les méthodes Bayesiennes.

Cela a été testé (Lemmon and Milinkovitch, 2002) en comparant les valeurs de support obtenues avec MetaPIGA aux probabilités postérieures obtenues avec le programme MrBayes (sur les mêmes données), qui utilise une méthode baysienne (voir section 4.3) : l'algorithme génétique métapopulationnel (MetaGA) a été appliqué dix fois, avec dix populations, et un arbre consensus a été calculé à partir des cent meilleurs arbres résultant. Les valeurs de support des branches semblent bien correspondre aux probabilités postérieures de MrBayes (figure 3.4).

Cependant, très peu de tests ont été effectuées, et sur un très petit nombre de données. Les résultats ne sont donc pas très consistants statistiquement.

Par ailleurs, de nombreuses études semblent démontrer que les probabilités postérieures estimées grâce à MrBayes sont faussées, et ont tendance à supporter de fausses branches en leur donnant une probabilité postérieure très élevée (voir section 4.3.1).

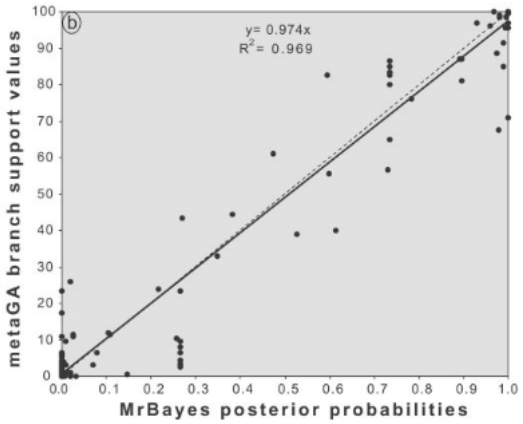


FIG. 3.4 – (Lemmon and Milinkovitch, 2002) Comparaison des valeurs de support des branches obtenues par MetaPIGA et des probabilités postérieures obtenues par MrBayes

3.6 Futur de MetaPIGA

Comme mentionné précédemment, MetaPIGA est encore en développement, et plusieurs outils doivent encore être implémentés.

De même, certains bugs sont encore présents, qui devraient être corrigés au fur et à mesure de leur découverte.

Futures implémentation

Un algorithme d'optimisation des paramètres permettant d'optimiser les longueurs de branches, les taux de substitution, ainsi que, s'ils sont définis, le paramètre de la distribution gamma et la proportion de sites invariants va être prochainement implémenté. Les paramètres pourront être optimisés soit uniquement à la fin de l'algorithme, soit régulièrement pendant la recherche de l'arbre phylogénétique de plus haute vraisemblance.

Un module permettant de déterminer quel modèle de substitution d'ADN utiliser va également être ajouté à MetaPIGA.

Le modèle de substitution le plus approprié varie fort d'un set de données à l'autre, et utiliser un modèle trop complexe quand ce n'est pas nécessaire peut mener à des problème d'over-fitting, en plus de ralentir l'algorithme.

Certains programmes d'ajustement de modèle existent déjà (exemple : ModelTest (Posada and Crandall, 1998)).

Une condition d'arrêt va être ajoutée aux trois options déjà disponibles. Lorsque cette option sera choisie, le programme fera des réplicats de la recherche afin de construire un arbre consensus, et s'arrêtera lorsque les valeurs de support des branches de cet arbre consensus se seront stabilisées.

La recombinaison des arbres sélectionnés pour former la nouvelle génération, pour l'algorithme génétique et l'algorithme génétique métapopulationnel va être implémentée.

Enfin, toute idée géniale survenant dans les prochains mois et applicable à la reconstruction d'arbres phylogénétique en utilisant le maximum de vraisemblance est susceptible d'être utilisée pour améliorer le programme.

Chapitre 4

Les « concurrents » de MetaPIGA

4.1 PAUP

Comme son nom l'indique, PAUP - *Phylogeny Analysis Using Parsimony* - (Swofford, 2003) implémentait au départ une méthode de parcimonie. De nombreux outils et méthodes ont été ajoutés depuis la première version de PAUP, et les méthodes de distances et de maximum de vraisemblance sont maintenant également disponibles.

Les méthodes UPGMA et NJ sont toutes deux implémentées, et les résultats peuvent servir d'arbres de départ pour les méthodes de parcimonie et de maximum de vraisemblance.

Des méthodes exactes (donnant la solution optimale) sont disponibles pour la méthode de parcimonie (recherche exhaustive et Branch and Bound (Hendy and Penny, 1982)), et un algorithme de Hill-Climbing peut être utilisé, à la fois avec le critère de parcimonie et de maximum de vraisemblance. PAUP implémente de plus une méthode de quartet-puzzling (voir section 2.7)

PAUP permet l'utilisation de nombreux modèles de substitutions de l'ADN, incluant la distribution gamma. Il peut également construire des arbres à partir de séquences ARN ou protéiques et implémente de nombreux outils permettant entre autres d'optimiser les longueurs de branches, manipuler les arbres phylogénétiques (enracinement, exportation, comparaison ou combinaison de plusieurs arbres...) ainsi que la méthode de bootstrap proposé par Felsenstein (Felsenstein, 1985), qui permet d'évaluer la confiance que l'on peut avoir en chaque branche d'un arbre consensus construit à partir des résultats de nombreuses répétitions d'un algorithme d'inférence phylogénétique.

Ce programme fut, avec PHYLIP, le plus utilisé pour l'inférence phylogénétique jusqu'à il y a quelques années.

4.1.1 Branch and Bound

L'algorithme de Branch and Bound (B&B) permet de trouver la solution optimale sans devoir tester toutes les topologies possibles. Il est donc plus rapide que la recherche exhaustive, sans perte de qualité des résultats.

Un arbre de recherche est construit par cet algorithme de la manière suivante : la racine contient l'unique topologie possible pour trois taxa pris dans les données considérées (arbre non raciné) ; les noeuds fils de cette racine contiendront eux les trois topologies possibles obtenues en ajoutant l'un des taxa restant sur l'une des trois branches de la topologie à trois taxa. Et ainsi de suite, les fils d'un noeud seront le résultat des différentes possibilités d'additions d'un taxa à l'arbre. L'arbre de recherche est construit en commençant par la profondeur, c'est-à-dire qu'à partir de la racine, on va construire un des fils de celle-ci, puis un des petit-fils, et ainsi de suite jusqu'à ce que tous les taxa ait été ajoutés. On reviendra ensuite au père du dernier arbre considéré pour construire une nouvelle solution possible.

Chaque topologie, partielle ou contenant tous les taxa, est évaluée selon le principe de parcimonie dès sa construction. De ce fait, si une topologie partielle est moins parcimonieuse que la meilleure

solution actuelle, il n'est pas nécessaire de continuer à ajouter des taxa à cette solution (car cela ne peut qu'augmenter le nombre d'événements évolutifs nécessaires pour expliquer les données). Ainsi, tous les noeuds fils de cette solution partielle ne seront pas exploré. Le B&B évite donc de grandes parties de l'espace de recherche, tout en étant certain de connaître l'arbre le plus parcimonieux à la fin de l'algorithme.

4.1.2 Hill-Climbing

L'algorithme de HC implémenté dans PAUP peut être utilisé avec les critères de parcimonie ou de maximum de vraisemblance.

L'algorithme démarre avec un arbre qui peut avoir été fourni par l'utilisateur, ou construit par le programme, soit grâce à une méthode de distances, soit par *Stepwise Addition*, c'est-à-dire qu'un arbre non raciné contenant seulement trois taxa est construit (une seule topologie possible), auquel on ajoute les taxa restant un par un. A chaque addition, les différentes topologies possibles sont testées, et celle qui satisfait le mieux le critère choisi est acceptée.

Une fois cet arbre de départ obtenu, il subit des réarrangements topologiques (NNI, TBR et SPR) qui sont acceptés s'ils améliorent l'arbre.

4.1.3 Bootstrap

Le bootstrap (Felsenstein, 1985) permet d'attribuer des valeurs de support aux branches d'un arbre phylogénétique consensus construit de la manière suivante :

Les données sont constitués de t séquences contenant chacune n caractères (ou sites). On a donc une matrice $t \times n$ de sites. Les colonnes de cette matrice vont être échantillonnées, avec remise, pour créer une nouvelle matrice $t \times n$ dont les lignes représenteront toujours t taxa différents, mais certaines colonnes de la matrice de départ pourront avoir été tirées plusieurs fois, et d'autres n'être pas représentées. Un arbre phylogénétique sera alors inféré à partir de cette nouvelle matrice, et cette opération est répétée r fois.

A la fin, r arbres auront donc été inférés, à partir de différents échantillons des caractères utilisés. Un arbre consensus est alors construit à partir de ces arbres (une branche particulière est retenue lorsqu'elle est majoritairement représentée parmi les r arbres) et à chaque branche est associée une valeur de support correspondant à la proportion d'arbres présentant cette branche.

Plusieurs hypothèses a priori sont faites lorsque l'on réalise un bootstrap, dont la plus importante est que les sites sont indépendants et identiquement distribués.

Cette hypothèse n'est pas réaliste. En effet, certains caractères ne sont pas indépendants. Par exemple, dans le cas de sites nucléotidiques, les nucléotides composant un codon ne sont pas indépendants. De plus, tous les sites n'ont pas une distribution identique, comme vu section 2.2.6.

Les valeurs de support calculées par bootstrapping doivent donc être considérées avec prudence.

4.2 PHYLIP

PHYLIP - *PHYLogeny Inference Package* - (Felsenstein, 2005) fut également très utilisé jusqu'à il y a quelques années.

Cet ensemble de programmes permet, tout comme PAUP, d'inférer des arbres phylogénétiques à partir de séquences ADN, ARN ou protéiques, mais également de fréquences de gènes, sites de restriction, caractères discrets ou continus..., grâce à des méthodes de distances, de parcimonie ou de maximum de vraisemblance. Les algorithmes utilisés sont forts proches des algorithmes implémentés par PAUP (B&B, HC avec Stepwise Addition et réarrangements topologiques)

PHYLIP supporte également plusieurs modèles de substitution de l'ADN (quoique moins que PAUP) et fournit plusieurs outils permettant de manipuler des arbres phylogénétiques (édition, enracinement, affichage...), de comparer plusieurs arbres ou de calculer des distances entre arbres, de faire du bootstrap, calculer un arbre consensus...

4.3 MrBayes

Ce programme a littéralement détrôné PHYLIP et PAUP et est aujourd'hui le plus utilisé.

MrBayes (Huelsenbeck and Ronquist, 2001; Ronquist and Huelsenbeck, 2003) utilise une méthode Bayesienne (voir section 2.6), avec une variante des MCMC : une Metropolis-coupled MCMC ((MC^3)).

Cet algorithme va construire n chaînes de Markov en parallèle, dont $n-1$ sont « chauffées ». Les chaînes chauffées ont une distribution des probabilités postérieures $P(H|D)^\beta$, chaque chaîne i (avec i allant de 2 à n) ayant un facteur β (la chaleur) différent, calculé comme suit :

$$\beta = \frac{1}{1+(i-1)T} \tag{4.1}$$

T étant un paramètre réglé par l'utilisateur.

Cela a pour effet de descendre les pics et remplir les vallées de la distribution des probabilités postérieures. Ainsi, plus une chaîne est chauffé, plus elle peut traverser facilement les vallées.

Seule la chaîne non chauffée est utilisée pour estimer les probabilités postérieures, mais, à chaque étape de l'algorithme (après l'acceptation ou non d'un nouvel état dans la chaîne), deux chaînes j et k parmi les n sont tirées au hasard, et ces deux chaînes ont une probabilité R d'échanger leur état (c'est-à-dire le dernier arbre qu'elles ont accepté), avec

$$R = min\left(1, \frac{P(\Psi_k|D)^{\beta_j} P(\Psi_j|D)^{\beta_k}}{P(\Psi_j|D)^{\beta_j} P(\Psi_k|D)^{\beta_k}}\right) \tag{4.2}$$

De cette manière, la chaîne non chauffée explore localement la distribution des probabilités postérieures, et change régulièrement de région, grâce à l'échange de son état avec celui d'une chaîne qui explore des régions beaucoup plus vastes.

MrBayes permet l'utilisation de nombreux types de données différents (nucléotides, protéines, sites de restriction, données morphologiques...) et plusieurs modèles stochastiques sont disponibles pour chacun d'eux.

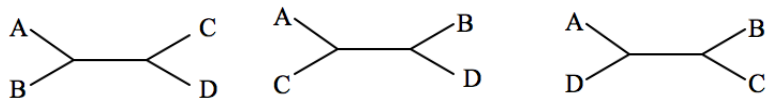
4.3.1 Surestimation des probabilités postérieures

D'après (Tierney, 1994), l'échantillonage d'une MCMC donne une approximation acceptable des probabilités postérieures des entités formant la chaîne.

Cependant, de nombreuses études mettent en doute la fiabilité des probabilités postérieures des arbres estimées par MrBayes.

Un set de données simulées a par exemple été construit en joignant les séquences de trois « gènes » pour quatre taxa (Suzuki *et al.*, 2002). Ces trois gènes ont été simulés à l'aide du programme SEQGEN (Rambaut and Grass, 1997), qui génère des séquences respectant une phylogénie donnée et selon un modèle de substitution d'ADN choisi par l'utilisateur. Chacun de ces trois gènes a été généré suivant un des trois arbres non racinés possibles pour quatre taxa (figure 4.1). La reconstruction de l'arbre phylogénétique des quatre taxa à partir des séquences jointes des trois gènes devrait donc donner chacune des topologies une fois sur trois en moyenne, et ces arbres ne devraient pas être supportés par de hautes valeurs de bootstrap ou de hautes probabilités postérieures. Deux bootstraps (avec le critère de maximum de vraisemblance et avec une méthode des distances) ont donc été réalisés, ainsi qu'une estimation des probabilités postérieures avec MrBayes (50 répétitions).

Dans le cas du bootstrap, moins de 5% des cas présentaient une probabilité supérieure à 95%, que les arbres ait été construits avec une méthode de distances ou le maximum de vraisemblance. Le bootstrap ne surestime donc pas les valeurs de support et semble même les sous-estimer légèrement. Par contre, dans 42% des cas, les probabilités postérieures calculées avec MrBayes étaient supérieures à 95%, ce qui montre une surestimation de ces probabilités postérieures.



(a) Topologie 1 (b) Topologie 2 (c) Topologie 3

FIG. 4.1 – (Suzuki *et al.*, 2002) Arbres utilisés pour simuler les données avec SEQGEN.

Plusieurs autres études montrent que les probabilités postérieures obtenues par MrBayes supportent, avec une très haute probabilité, des branches erronées (Douady *et al.*, 2003; Cummings *et al.*, 2003; Erixon *et al.*, 2003; Jones, 2008), et que ces probabilités sont sujettes à de nombreux biais, provoqués notamment par un modèle de substitution de l'ADN mal choisi (Huelsenbeck and Rannala, 2004), ou les probabilités antérieures des longueurs de branches (Yang and Rannala, 2005; Kolaczkowski and Thornton, 2007).

4.4 GARLI

GARLI - *Genetic Algorithm for Rapid Likelihood Inference* - (Zwickl, 2006) est un programme d'inférence phylogénétique basé sur un algorithme génétique et s'inspirant du programme GAML (Lewis, 1998).

Il utilise le critère de maximum de vraisemblance et peut construire des arbres phylogénétiques à partir de séquences ADN, protéiques ou de codons. Pour chacun de ces types de séquences, il supporte plusieurs modèles d'évolution.

Tout comme MetaPIGA, les mutations affectant les individus de la population peuvent modifier la topologie de l'arbre, les longueurs de branches ou les paramètres du modèle de substitution. Les réarrangement topologiques autorisés sont les NNI et les SPR, et chaque réarrangement topologique est suivi d'une optimisation partielle des longueurs de branches.

4.5 Autres programmes d'inférence phylogénétique

Il existe actuellement des dizaines de programmes permettant de construire des arbres phylogénétiques. Sont listés ici quelques-uns des plus connus.

GAML (Lewis, 1998) utilise le critère de maximum de vraisemblance et est basé sur un algorithme génétique.

Tree-Puzzle (Schmidt *et al.*, 2002) utilise le maximum de vraisemblance et est basé sur le quartet-puzzling.

MEGA (Kumar *et al.*, 2008) peut utiliser les méthodes de distances ou la parcimonie.

DAMBE (Xia and Xie, 2001) utilise les méthodes de distances, de parcimonie et de maximum de vraisemblance et permet de faire du bootstrap.

DupTree (Wehe *et al.*, 2008) utilise le critère de parcimonie.

PhyML (Guindon and Gascuel, 2003) utilise le maximum de vraisemblance, basé sur un Hill-Climbing.

BAMBE (Simon and Larget, 2000) utilise une méthode Bayesienne.

Chapitre 5

Matériel et méthode

Pour chacune des quatre heuristiques de MetaPIGA, plusieurs paramètres ont été testés, et les résultats obtenus ont été comparés afin de déterminer quels paramètres sont les plus intéressants à utiliser. Les quatre heuristiques ont ensuite été comparées entre elles.

En cas d'utilisation de tests statistiques, une p-valeur inférieure à 0,05 est considéré comme en-deça du seuil de significativité.

Le même ordinateur, possédant 8 processeurs (2,66GHz) et 8,00 Gb de mémoire RAM, a été utilisé pour tous les tests.

5.1 Données utilisées

Tous les tests ont été faits sur un set de données simulé, à l'exception des comparaisons entre heuristiques, pour lesquelles, en plus des données simulées, un set de données réelles a été utilisé.

5.1.1 Données simulées

Le set de séquences ADN alignées utilisé pour l'optimisation des paramètre de MetaPIGA a été simulé avec le programme EvolveAGene (Hall, 2007).

L'utilisation de données simulées présente plusieurs avantages : on est certain que l'alignement est correct et l'on dispose d'un point de comparaison des résultats parfait puisque l'on connaît le « vrai » arbre phylogénétique associé aux séquences.

Les inconvénients sont bien sûr le manque de réalisme de ces données, d'une part du fait même de leur simulation (quel que soit le modèle utilisé, l'évolution naturelle des séquences est plus complexe), d'autre part parce que les biais dus aux possibles erreurs d'alignement ne sont absolument pas pris en compte.

EvolveAGene

EvolveAGene est un programme qui permet de simuler l'évolution d'une séquence ADN le long d'un arbre phylogénétique.

Contrairement à de nombreux autres programmes de ce type (comme le très utilisé SEQGEN (Rambaut and Grass, 1997)), la simulation ne se fait pas selon un modèle de substitution de l'ADN choisi par l'utilisateur mais par un processus de mutations et sélections.

Une séquence ADN, avec un nombre de caractères multiple de 3 (le processus de mutations/sélections étant basé sur les codons), doit être fournie au programme. Cette séquence va alors évoluer le long d'un arbre phylogénétique, dont la topologie peut être fournie par l'utilisateur ou initialisée au hasard par le programme, et dont les longueurs de branches sont initialisées au hasard entre zéro et deux fois une longueur moyenne déterminée par l'utilisateur. A chaque étape, dont le nombre dépend

des longueurs de branches, une mutation est faite sur la séquence (cela peut être une substitution, une insertion ou une délétion), et cette mutation a une certaine probabilité d'être acceptée (sélectionnée). Selon le type de mutation, cette probabilité sera différente : une mutation qui résulte en le remplacement d'un codon correspondant à un acide aminé en un codon « stop » ne sera jamais acceptée, tandis que les probabilités d'accepter une mutation silencieuse, une mutation non-silencieuse, une insertion ou une délétion sont toutes les quatre données par l'utilisateur (par défaut, une mutation silencieuse est toujours acceptée, et les autres probabilités sont basées sur des valeurs provenant de la littérature).

Le set de données utilisé a été simulé à partir d'une séquence de 1500 caractères, avec les paramètres suivants :

Nombre de taxa	20
Arbre	random
Longueur de branche moyenne	0,05
P(substitution silencieuse)	1,0
P(substitution non silencieuse)	0,016
P(insertion)	0,1
P(délétion)	0.025

Les probabilités de mutations correspondent aux valeurs par défaut. Le set de données résultant de cette simulation comprend 20 séquences alignées de 1575 nucléotides chacune.

5.1.2 Données réelles

Le set de données réel provient d'une étude de la phylogénie des Ranidae (Bossuyt *et al.*, 2006) et comprend 111 séquences de 3679 caractères. En plus d'être réel, il est donc aussi beaucoup plus grand que le set de données simulées, et correspond beaucoup plus aux tailles de données pour lesquelles MetaPIGA a été développé.

5.2 Méthode

La première et la plus importante partie de ce travail concerne le test des paramètres. Pour les quatre heuristique implémentées dans MetaPIGA, plusieurs paramètre ont été testés à partir des données simulées.

Les quatre heuristiques ont ensuite été comparées. Chacune a été paramétrée selon les résultats obtenus dans la première partie du travail, et utilisée à la fois sur les données simulées et sur le set de données réel, cette dernière comparaison étant faite dans le but notamment de voir à quel point la taille des données influence les résultat. En particulier, on s'attend à ce que l'algorithme génétique et l'algorithme génétique métapopulationnel ne donnent pas des résultats particulièrement meilleurs que le Hill-Climbing et le Recuit Simulé pour le set de données simulé, mais qu'ils soient beaucoup plus intéressants pour le set de données réelles.

Petite remarque sémantique

La vraisemblance des arbres phylogénétique, même lorsque ces arbres sont extrêmement proches du vrai arbre phylogénétique, est très petite, et diminue avec la taille des données. A titre d'exemple, la vraisemblance du vrai arbre des données simulées utilisées ici est inférieure à 10^{-5385}.

Par conséquent, la vraisemblance des arbres explorés avec MetaPIGA (et la plupart des programmes d'inférence phylogénétique) n'est pas donnée telle quelle, mais sous forme de logarithme népérien de la vraisemblance ($\ln(L)$), ce qui résulte en des nombres plus faciles à traiter et à comparer (ne fût-ce que parce que beaucoup de programmes et langages informatiques ne peuvent gérer directement des nombres aussi petits que 10^{-5385}).

Aussi, dans la suite de ce travail, lorsque je parlerai de comparaison de vraisemblances obtenues, ou simplement de vraisemblance des arbres obtenus, il s'agira d'un abus de langage destiné à alléger le texte, et je me référerai en réalité à $-\ln(L)$.

5.2.1 Optimisation des paramètres

Lors de l'optimisation des paramètres, 50 arbres génétiques ont été construits avec MetaPIGA pour chaque option testée.

Tous les tests ont été faits avec le modèle de JC, sans proportion d'invariants, ni distribution gamma. La condition d'arrêt utilisée est un arrêt automatique après 300 itérations de l'algorithme sans amélioration de la vraisemblance. Aucun outgroup ni set de charactères n'ont été défini et les gaps sont traités comme des N.

Par défaut, l'arbre (ou les arbres) de départ utilisé(s) est (sont) construit(s) par LNJ avec un pourcentage de 10%, les opérateurs utilisés sont le NNI, TXS (de 3 taxa), STS (de 2 sous-arbres), SPR, TBR et BLMINT, et ces opérateurs sont sélectionnés au hasard.

Pour l'algorithme de SA, le cooling schedule par défaut est Lundy, avec une diminution de la température toutes les 10 améliorations ou toutes les 100 diminutions de la vraisemblance. La température est remise à son niveau initial tous les 300 décréments de température. Les paramètres α et c sont deux tous mis à 0,5 (valeurs tirées de (Salter and Pearl, 2001)). Si nécessaire, les probabilités initiale et finale d'acceptance sont mises par défaut à 70% et 1% respectivement.

Les paramètres par défaut du GA sont : 8 individus dans la population, une sélection par amélioration, et les opérateurs sont appliqués séparément à chaque individu.

Quant au MetaGA, il utilise par défaut 4 populations de 4 individus chacune, une sélection par amélioration, et les opérateurs sont appliqués séparément à chaque individu. Le consensus est de type stochastique, les opérateurs sont sélectionnés de manière supervisée par rapport aux branches consensus et la tolérance est de 5%.

Le HC n'a aucun paramètre qui lui soit propre.

Paramètres testés sur toutes les heuristiques

- **Arbre de départ** NJ et LNJ avec un pourcentage de 10, 30, 50, 75 ou 100%.
- **Utilisation de l'opérateur BLM** activation ou non de l'opérateur BLM. Ce test visait à déterminer si la modification des longueurs des branches terminales permet d'améliorer la vraisemblance de manière significative.
- **Sélection des opérateurs** au hasard ou selon des fréquences dynamiques (fréquences initiales de 15% pour STS, NNI, TBR et SPR et de 20% pour TXS et BLMINT).

Paramètres testés sur le SA

- **Cooling schedule** tests fait sur Lundy, Boltzman, Ratio à 99%, tangente et cosinus hyperbolique, diminutions linéaire, polynomiale et logarithmique transcendentale.
- **Décrément de température** tests avec décrément toutes les 20 itérations de l'algorithme, après 10 améliorations ou 100 diminutions de la vraisemblance (10S100F), après 20S100F, 40S100F, 10S75F, 10S50F et 10S30F.
- **Remise de la température à T_0** SA lancé soit avec aucune remise à T_0, soit avec remise lorsque la température atteint 0.001% de T_0, soit avec remise après 100, 200 ou 300 décréments de la température.
- **Calcul de ΔL** par « burn-in » ou en prenant 0,1% de la vraisemblance de l'arbre construit par NJ.

Paramètres testés sur le GA

- **Taille de la population** de 2 à 20 individus.
- **Type de sélection appliqué** tests avec sélection par rang, par tournoi, par amélioration, du meilleur et par remplacement avec une force de sélection de 10, 30, 50, 60, 70, 80, 90 et 100%.

Paramètres testés sur le CP

• **Nombre de populations** de 2 à 5 populations évoluant en parallèle.

• **Taille des populations** de 2 à 10 individus par population.

• **Type de consensus** strict et stochastique.

• **Sélection des opérateurs vis-à-vis du consensus** aveugle ou supervisée

5.2.2 Comparaison des heuristiques

Pour les données simulées, 50 réplicats ont été faits avec chaque heuristique. Par contre, le temps de calcul nécessaire avec les données réelles étant beaucoup plus grand, il n'a été possible que de faire 10 réplicats par heuristique. Les résultats sur les données réelles sont donc sujets à grande caution.

Le paramétrage du programme pour chacune des heuristiques dépendant en partie des résultats des tests sur les paramètres, les options choisies seront détaillées après présentation de ces résultats, section 6.2.

5.3 Remarques

Etant donnés les temps de calcul assez élevés nécessaires à MetaPIGA et le temps imparti pour réaliser ce travail, les analyses effectuées ne sont qu'une infime partie de ce qu'il faudrait faire pour vraiment optimiser les paramètres.

Premièrement, seules des données simulées ont été utilisées pour optimiser les paramètres, et il serait certainement intéressant de refaire ses analyses sur des données réelles, bien qu'alors on ne dispose plus du « vrai » arbre phylogénétique.

Ensuite, les performances de MetaPIGA, et en particulier le temps de calcul nécessaire, est fort susceptible de varier selon la taille des données utilisées (à la fois le nombre de taxa et la longueurs des séquences). Il faudrait donc, pour être complet, tester chacun des paramètres sur des données de différentes tailles, simulées et réelles.

De plus, chaque paramètre n'a été testé que 50 fois (voire moins pour les données réelles), ce qui n'est pas suffisant pour avoir des statistiques vraiment solides.

Par ailleurs, pour certains paramètres, seuls certains choix ont été testés, il est donc tout à fait possible qu'un choix non testé ici se révèle par la suite meilleur que tous les autres.

Enfin, et c'est le point le plus important, MetaPIGA a été développé pour pouvoir inférer des arbres génétiques à partir de grands sets de données (contenant plusieurs dizaines, voire centaines de taxa). En particulier, les algorithmes génétiques (incluant metaGA) sont particulièrement efficaces sur de grandes phylogénies par rapport au HC ou au SA, mais pour de petits sets de données, ces deux dernières heuristiques suffisent amplement. De ce fait, tester les paramètres sur un petit set de données peut donner des informations utiles mais il faut garder à l'esprit que MetaPIGA est destiné à gérer des données bien plus volumineuses, et les résultats obtenus sur ce set de données réduit ne refléteront donc pas les véritables capacités du programme.

Ce travail est donc à considérer comme une étude préliminaires des performances de certains paramètres de MetaPIGA. Il reste de nombreuses choses à faire et à tester avant que ce programme ne soit réellement le plus efficace et performant possible.

Chapitre 6

Résultats

6.1 Optimisation des paramètres

6.1.1 Paramètres testés sur toutes les heuristiques

Les paramètres testés sur toutes les heuristiques sont les arbres de départ, l'utilisation de l'opérateur BLM et la sélection des opérateurs.

Arbres de départ

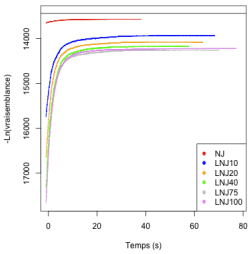
La figure 6.1 montre l'évolution de la vraisemblance en fonction du temps (moyenne sur les 50 réplicats) obtenue avec différents arbres de départ, pour les quatre heuristiques. On voit que lorsque l'on commence avec un arbre construit par Neighbour-Joining, on obtient les meilleurs résultats en terme de vraisemblance des arbres. Avec des arbres construit par Loose Neighbour-Joining, l'augmentation de la vraisemblance est plus importante, mais l'on atteint de moins bon résultats. Et plus le pourcentage du LNJ est grand, moins bons sont ces résultats.

Le temps moyen pour obtenir un arbre, ainsi que la vraisemblance finale sont montrés sur la figure 6.2, et les valeurs données dans la table 6.1.

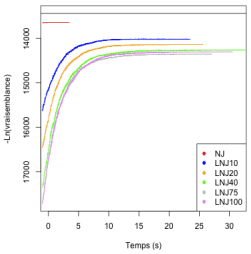
Avec un arbre NJ, l'écart-type sur la vraisemblance est extrêmement faible. L'algorithme de NJ donnant un très bon arbre, en particulier pour de petits sets de données, ce n'est pas particulièrement étonnant. Cependant, en ce qui concerne le SA, le GA et le CP, on se serait attendu à ce qu'ils permettent, de temps en temps, de sortir de l'optimum local près duquel se trouve l'arbre NJ, et donc à trouver un écart-type sur la vraisemblance plus important. Il est fort probable que ce soit la taille réduite du set de données qui entraîne ces résultat. En effet, le NJ donne de bien meilleurs résultats pour de petites phylogénies que pour des grandes. De ce fait, pour le GA et le CP, les individus sélectionnés seront très majoritairement des arbres gardant la topologie de l'arbre de départ (qui est probablement la « vraie topologie », ou très proche de celle-ci) avec une optimisation des branches grâce à l'opérateur BLMINT. Quant au SA, il acceptera très rarement les changements topologiques importants car ceux-ci feront descendre drastiquement la vraisemblance, et il aura donc aussi tendance à rester sur ce même pic. Cela explique aussi le temps de recherche réduit : rapidement, l'optimum local est atteint ou presque, puisque l'arbre de départ est déjà très proche de cet optimum, et il devient tellement rare de trouver une modification qui permette d'augmenter la vraisemblance que la condition d'arrêt est atteinte.

Lorsque l'arbre de départ est construit avec le LNJ, les résultats sont moins bons, et l'écart-type plus important. La stochasticité introduite dans la construction de l'arbre de départ permet d'accepter plus souvent des réarrangements topologiques, et permet éventuellement de sortir de l'optimum local où mène l'algorithme lorsque l'on commence avec des arbres NJ. Elle rallonge également le temps de recherche puisqu'il y a plus de changements susceptibles de faire augmenter la vraisemblance.

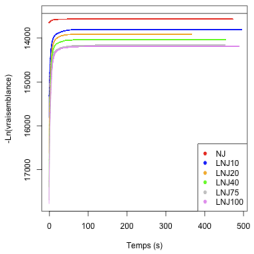
Ces résultats pourraient inciter à toujours utiliser l'arbre NJ pour démarrer une heuristique. Il faut cependant nuancer ceux-ci. Pour la construction de grandes phylogénies, utiliser l'arbre NJ pourrait bien bloquer l'heuristique dans un optimum local, qui ne soit plus aussi bon que dans le cas de phylogénies avec peu de taxa, comme c'est le cas ici.



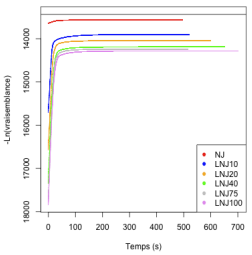
(a) Hill-Climbing



(b) Recuit simulé

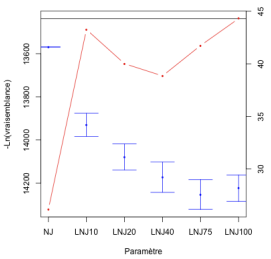


(c) Algorithme Génétique

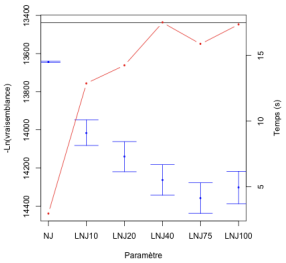


(d) Consensus Pruning

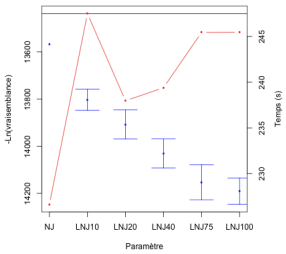
FIG. 6.1 – Evolution de $-\ln(L)$ avec différents arbres de départ (moyenne sur 50 réplicats) pour (a) HC, (b) SA, (c) GA et (d) MetaGA. Les nombres mentionnés après « LNJ » correspondent au pourcentage du LNJ. Les droites horizontales tracées en noir correspondent à $-\ln(L)$ du vrai arbre phylogénétique.



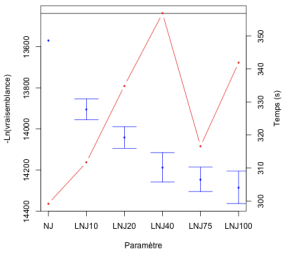
(a) Hill-Climbing



(b) Recuit simulé



(c) Algorithme Génétique



(d) Consensus Pruning

FIG. 6.2 – Moyenne de $-\ln(L)$ (en bleu - axe de gauche) avec l'écart-type correspondant, et moyenne du temps de calcul (en rouge - axe de droite) pour (a) HC, (b) SA, (c) GA et (d) MetaGA. Les nombres mentionnés après « LNJ » correspondent au pourcentage du LNJ. Les droites horizontales tracées en noir correspondent à $-\ln(L)$ du vrai arbre phylogénétique.

		NJ	LNJ10	LNJ20	LNJ40	LNJ75	LNJ100
HC	$-\ln(L)$	13569.09	13929.95	14078.45	14172.35	14252.47	14223.61
	écart-type	1.49	107.21	121.02	140.41	138.44	122.27
HC	Temps (s)	26.15	43.26	40.01	38.85	41.74	44.30
	écart-type	6.80	12.18	9.28	9.52	13.31	10.79
SA	$-\ln(L)$	13643.45	14014.92	14140.73	14262.71	14357.87	14302.72
	écart-type	7.50	134.12	157.54	159.98	161.01	170.07
SA	Temps (s)	2.97	12.87	14.24	17.49	15.86	17.34
	écart-type	0.44	4.25	3.93	5.60	4.87	4.99
GA	$-\ln(L)$	13567.90	13803.19	13907.55	14030.78	14153.40	14191.36
	écart-type	0.02	90.34	124.11	123.87	149.98	111.65
GA	Temps (s)	226.63	247.50	237.97	239.39	245.45	245.47
	écart-type	74.05	66.54	63.53	69.58	63.28	82.61
CP	$-\ln(L)$	13568.00	13904.48	14041.46	14186.75	14245.29	14284.39
	écart-type	0.12	100.92	104.85	143.15	119.28	158.24
CP	Temps (s)	299.21	311.70	334.94	356.82	316.61	341.91
	écart-type	86.07	97.69	86.64	109.53	85.36	105.61

TAB. 6.1 – Moyennes et écarts-type de $-\ln(L)$ des arbres obtenus avec les quatre heuristiques (L : vraisemblance), et moyennes et écarts-type du temps de recherche, pour différents arbres de départ. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

Utilisation de l'opérateur BLM

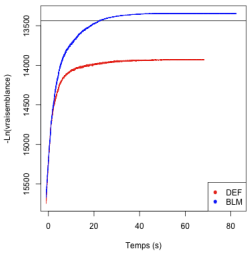
L'évolution de la vraisemblance lorsque l'opérateur BLM est activé ou non est montrée figure 6.3. Lorsqu'il est utilisé, on constate que la vraisemblance moyenne des arbres inférés dépasse rapidement, sauf dans le cas du SA, la vraisemblance du vrai arbre phylogénétique. La longueur des branches terminales influence donc beaucoup sur la vraisemblance de l'arbre, mais semble sujette à l'over-fitting.

L'activation de l'opérateur BLM rallonge le temps de recherche, mais semble particulièrement réduire l'écart-type sur la vraisemblance, à l'exception à nouveau du SA (table 6.2).

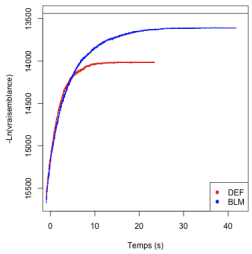
La raison pour laquelle le SA se comporte différemment des trois autres heuristiques n'est pas évidente. Une hypothèse serait que cette heuristique reste bloquée dans un optimum local mais que cet optimum local serait très différent d'un réplicat à l'autre, tandis que le HC, le GA et le CP retomberait souvent sur le même optimum, ou du moins sur des optima de semblable vraisemblance, fort proche de l'optimum global. Cela serait dû au fait que ces trois dernières heuristiques tendent uniquement à monter (de par la nature même du HC et la sélection appliquée aux algorithmes génétiques : sélection par amélioration), tandis que le SA accepte de temps en temps des solutions moins bonnes, et éventuellement beaucoup moins bonnes.

La tendance à l'over-fitting lorsque l'on utilise l'opérateur BLM pourrait poser problème et bloquer l'heuristique dans un mauvais optimum local. En effet, si l'optimisation de la longueur des branches terminales permet d'augmenter énormément la vraisemblance comme cela semble être le cas, on peut imaginer que le programme donne comme résultat un arbre avec une topologie fausse car l'optimisation de ses longueurs de branches terminales lui a conféré une telle vraisemblance que l'heuristique n'a pu accepter comme meilleur un arbre dont la topologie est plus proche de la réalité mais dont les branches terminales ne sont pas optimisées. Seul le CP devrait pouvoir échapper à ce genre de biais car la probabilité que plusieurs populations évoluant en parallèle se retrouvent bloquée dans le même optimum local est extrêmement faible, et l'algorithme continuerait donc à tourner jusqu'à sortir de cette mauvaise topologie.

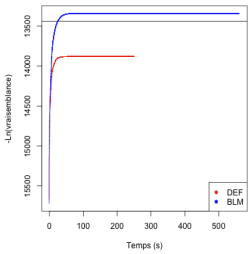
Il vaut donc sans doute mieux ne pas utliser cet opérateur (ou à une fréquence très réduite) et optimiser les longueurs de branches une fois la condition d'arrêt atteinte.



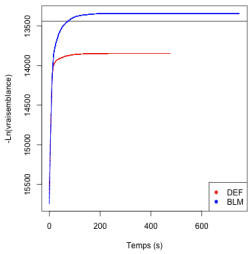
(a) Hill-Climbing



(b) Recuit simulé



(c) Algorithme Génétique



(d) Consensus Pruning

FIG. 6.3 – Evolution de $-\ln(L)$ avec et sans activation de l'opérateur BLM (moyenne sur 50 réplicats) pour (a) HC, (b) SA, (c) GA et (d) MetaGA. *DEF* : opérateurs par défaut, donc sans BLM, *BLM* : opérateur BLM activé. Les droites horizontales tracées en noir correspondent à $-\ln(L)$ du vrai arbre phylogénétique.

	HC		SA	
	sans BLM	avec BLM	sans BLM	avec BLM
$-\ln(L)$	13929.95	13344.90	14014.92	13610.90
écart-type	107.21	5.91	134.12	163.40
Temps (s)	43.26	57.85	12.87	21.57
écart-type	12.18	10.80	4.25	7.92
	GA		CP	
	sans BLM	avec BLM	sans BLM	avec BLM
$-\ln(L)$	13878.26	13339.97	13845.47	13340.33
écart-type	170.39	0.06	121.95	0.44
Temps (s)	97.73	319.01	262.22	464.20
écart-type	66.05	85.87	82.34	128.42

TAB. 6.2 – Moyenne et écart-type de $-\ln(L)$ des arbres obtenus avec les différents heuristiques (L : vraisemblance), et moyenne et écart-type du temps de recherche, avec ou sans activation de l'opérateur BLM. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

Opérateurs sélectionnés au hasard ou selon des fréquences dynamiques

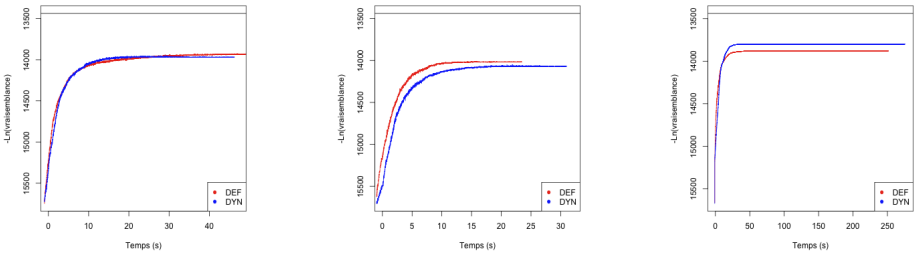
La figure 6.4 montre l'évolution de la vraisemblance lorsque les opérateurs sont sélectionnés au hasard et lorsqu'ils sont sélectionnés selon des fréquences évoluant dynamiquement.

A l'exception du GA, les heuristiques ne semblent pas donner de meilleurs résultats lorsque les fréquences des opérateurs sont dynamiques (moyennes et écart-type table 6.3). Cela est confirmé par un test de Mann-Whitney, qui permet de comparer les moyennes de deux distributions. On obtient des p-valeurs de 0.647, 0.063 et 0.414 pour la comparaison des moyennes obtenues avec les deux options pour le HC, le SA et le CP respectivement et une valeur de 0.005 pour le GA. Les algorithmes HC et SA sont probablement trop rapides pour que les fréquences puissent s'ajuster de manière efficace pour un set de données de cette taille, quant au CP, les fréquences des opérateurs devraient sans doute être ajustées pour chaque population séparément étant donné qu'elles évoluent

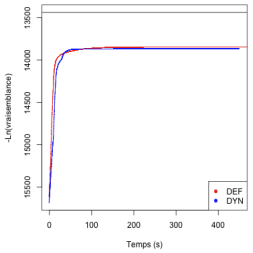
en parallèle donc pas nécessairement au même rythme et de la même manière.

Au niveau du temps de recherche, on n'observe pas une tendance nette pour toutes les heuristiques. Le GA semble particulièrement ralenti par les fréquences dynamiques, le HC également quoique de manière moins nette, tandis que le CP et le SA semblent prendre le même temps dans les deux cas. Cependant, les écarts-type sur les temps de recherche sont très importants et l'on peut donc difficilement tirer des conclusions.

La sélection des opérateurs selon des fréquences changeant selon l'efficacité de ceux-ci ne semble donc intéressante que pour le GA, et cela se fait au prix d'un allongement du temps de calcul.



(a) Hill-Climbing (b) Recuit simulé (c) Algorithme Génétique



(d) Consensus Pruning

FIG. 6.4 – Evolution de $-\ln(L)$ avec opérateurs sélectionnés au hasard et selon des fréquences dynamques (moyenne sur 50 réplicats) pour (a) HC, (b) SA, (c) GA et (d) MetaGA. *DEF* : opérateurs par défaut, donc sélection au hasard, *DYN* : opérateurs sélectionnés selon des fréquences dynamiques. Les droites horizontales tracées en noir correspondent à $-\ln(L)$ du vrai arbre phylogénétique.

	HC		SA	
	Au hasard	Freq. dynamiques	Au hasard	Freq. dynamiques
$-\ln(L)$	13929.95	13969.48	14014.92	14072.50
écart-type	107.21	202.85	134.12	172.82
Temps (s)	43.26	25.70	12.87	14.27
écart-type	12.18	11.25	4.25	5.98
	GA		CP	
	Au hasard	Freq. dynamiques	Au hasard	Freq. dynamiques
$-\ln(L)$	13878.26	13799.11	13845.47	13867.93
écart-type	170.39	94.75	121.95	121.67
Temps (s)	97.73	166.26	262.22	273.87
écart-type	66.05	46.11	82.34	91.60

TAB. 6.3 – Moyenne et écart-type de $-\ln(L)$ des arbres obtenus avec les différents heuristiques (L : vraisemblance), et moyenne et écart-type du temps de recherche, avec opérateurs sélectionnés au hasard et selon des fréquences dynamques. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

6.1.2 Paramètres testés sur le SA

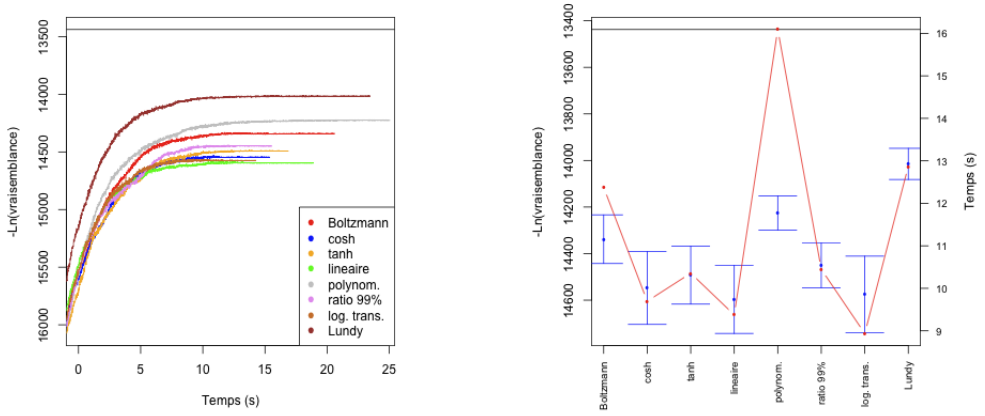
La manière dont la température diminue (=*cooling schedule*), ainsi que la fréquence à laquelle elle diminue et à laquelle elle est remise à la température initiale ont été testées, ainsi que les deux possibilités de calcul de ΔL.

Cooling schedule

Huit des quatorze cooling schedules disponibles ont été testés. Les résultats sont montrés figure 6.5(a) pour l'évolution de la vraisemblance et figure 6.5(b) et table 6.4 pour les moyennes des vraisemblances finales et du temps de recherche.

On voit clairement que le cooling schedule Lundy semble supérieur aux autres en terme de vraisemblance finale. Vient ensuite la diminution polynomiale, puis Boltzmann, et les autres cooling schedules semblent donner à peu près les mêmes résultats. La table 6.5 donne les p-valeurs d'un test de Mann-Whitney effectué sur chaque paire de cooling schedules testés possible. Et en effet, la vraisemblance moyenne obtenue avec Lundy est significativement supérieure aux vraisemblances moyennes obtenues avec les autres cooling schedules, la diminution polynomiale donne également une vraisemblance significativement supérieure à celles obtenues avec tous les cooling schedules sauf Lundy, même chose pour Boltzman (supérieure pour tous sauf Lundy et diminution polynomiale). Les autres cooling schedules ne donnent pas de résultats significativement différents l'un de l'autre, à l'exception de la diminution linéaire qui semble donner des résultats significativement inférieurs à ceux obtenus avec le ratio à 99%, avec une p-valeur de 0,011 (ce qui est beaucoup moins significatif que les différences avec Lundy, Boltzmann et la diminution polynomiale).

Lundy semble donc le meilleur cooling schedule parmi ceux testés. A noter que la diminution polynomiale, si elle donne de bons résultat, est aussi le cooling schedule qui allonge le plus le temps de recherche de l'arbre.



(a) (b)

FIG. 6.5 – (a) Evolution de $-\ln(L)$ et (b) moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus avec différents cooling schedules. Les droites horizontales tracées en noir correspondent à $-\ln(L)$ du vrai arbre phylogénétique.

	Boltz.	cosh	tanh	linéaire	polynom.	ratio 99%	log. trans.	Lundy
$-\ln(L)$	14338.04	14547.53	14492.27	14596.64	14225.71	14450.27	14575.44	14014.92
écart-type	208.38	312.39	248.62	292.00	146.79	193.15	329.51	134.12
Temps (s)	12.37	9.70	10.35	9.38	16.09	10.45	8.94	12.87
écart-type	3.82	2.64	3.19	3.48	4.08	2.64	2.67	4.25

TAB. 6.4 – Moyenne et écart-type de $-\ln(L)$ et du temps de calcul obtenus avec différents cooling schedules pour le SA. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

	cosh	tanh	linéaire	polynom.	ratio 99%	log. trans.	Lundy
Boltz.	2.1×10^{-4}	4.0×10^{-3}	9.8×10^{-06}	7.1×10^{-3}	7.6×10^{-3}	8.4×10^{-05}	1.5×10^{-12}
cosh		0.39	0.30	1.46×10^{-09}	0.15	0.74	6.7×10^{-16}
tanh			0.066	6.6×10^{-08}	0.56	0.31	2.2×10^{-16}
linéaire				6.2×10^{-11}	0.011	0.45	2.2×10^{-16}
polynom.					5.9×10^{-08}	3.8×10^{-10}	8.7×10^{-10}
ratio 99%						0.099	2.2×10^{-16}
log. trans.							$< 10^{-15}$

TAB. 6.5 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des $-\ln(L)$ pour différents cooling schedules.

Décrement de la température

La figure 6.6(a) montre l'évolution de la vraisemblance pour les diverses options testées concernant la diminution de la température. La figure 6.6(b) donne les moyennes des vraisemblances finales obtenues, ainsi que le temps de recherche moyen de l'algorithme, tandis que la table 6.6 donne les valeurs de ces moyennes, ainsi que les écarts-type. Les différentes options semblent donner les mêmes résultats en terme de vraisemblance. Par contre, le temps de recherche est visiblement minimal lorsque la température est diminué toutes les 10 améliorations ou 100 diminutions de la vraisemblance (10S100F) et maximal lorsqu'elle est diminué toutes les 20 itérations de l'algorithme (20ST).

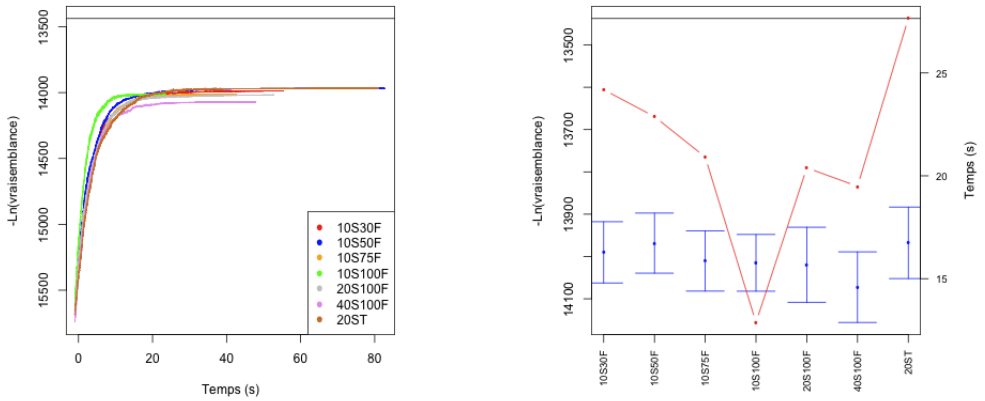
La table 6.7 donne les p-valeurs obtenues pour un test de Mann-Whitney effectué sur chaque paire possible de distribution de vraisemblances obtenues grâce aux différentes options de diminution de la température, et l'on constate qu'effectivement, aucune option ne donne de résultats très significativement supérieurs aux autres résultats. Seule la diminution après 40 améliorations ou 100 diminutions de la vraisemblance donne des résultats significativement inférieurs aux options 20ST, 10S30F et 10S50F.

On peut donc difficilement se baser sur la vraisemblance des arbres obtenus pour choisir une option. Seuls les temps de recherche pourraient donc nous permettre de faire un choix, lequel serait naturellement 10S100F.

Un argument basé sur le temps de recherche n'est cependant pas très solide, étant donné les écarts-type importants observés sur ceux-ci.

	10S30F	10S50F	10S70F	10S100F	20S100F	40S100F	20ST
$-\ln(L)$	13990.23	13968.40	14010.58	14014.92	14019.61	14072.52	13967.77
écart-type	144.90	142.04	142.30	134.12	177.51	166.91	168.93
Temps (s)	24.19	22.88	20.91	12.87	20.39	19.49	27.65
écart-type	10.72	12.66	7.48	4.25	9.18	8.09	13.38

TAB. 6.6 – Moyenne et écart-type de $-\ln(L)$ et du temps de calcul obtenus avec différentes options de diminution de la température du SA (50 réplicats). *20ST* : diminution de la température toutes les 20 itérations de l'algorithme. $xSyF$: diminution de la température toutes les x améliorations ou y diminutions de la vraisemblance. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)



(a) (b)

FIG. 6.6 – (a) Evolution de $-\ln(L)$ (moyenne sur les 50 réplicats) et (b) moyenne et écart-type de $-\ln(L)$ final (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus avec différentes options de diminution de la température. Les droites horizontales tracées en noir correspondent à $-\ln(L)$ du vrai arbre phylogénétique. *20ST* : diminution de la température toutes les 20 itérations de l'algorithme. *xSyF* : diminution de la température toutes les x améliorations ou y diminutions de la vraisemblance.

	10S50F	10S70F	10S100F	20S100F	40S100F	20ST
10S30F	0.62	0.49	0.27	0.29	0.0085	0.55
10S50F		0.22	0.078	0.11	0.00092	0.86
10S75F			0.54	0.66	0.053	0.38
10S100F				0.99	0.14	0.079
20S100F					0.23	0.15
40S100F						0.0027

TAB. 6.7 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des vraisemblances pour différentes options de diminution de la température du SA. *20ST* : diminution de la température toutes les 20 itérations de l'algorithme. *xSyF* : diminution de la température toutes les x améliorations ou y diminutions de la vraisemblance.

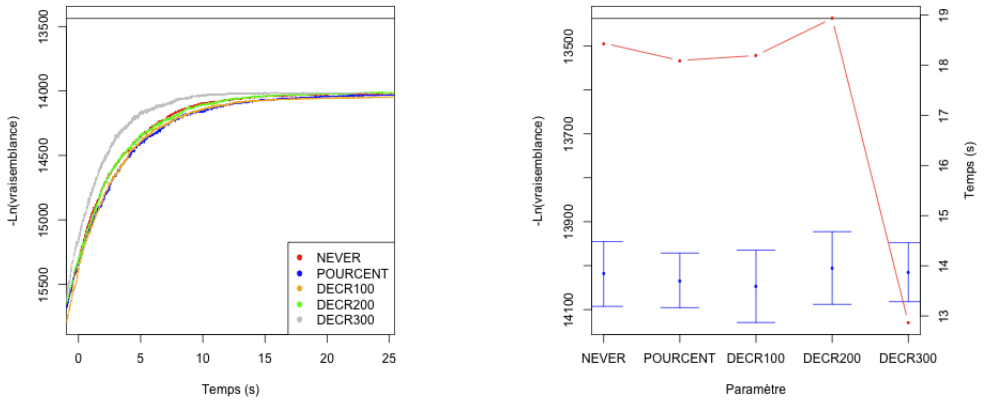
Remise de la température à T_0

Les résultats obtenus pour différentes options de remise de la température à la température initiale (T_0) sont montrés figure 6.7.

Les vraisemblances finales obtenues semblent être équivalentes quelle que soit l'option choisie. Par contre, le temps de recherche est moins grand lorsque l'on remet la température à T_0 après 300 décréments de la température que pour toutes les autres options (table 6.8).

Si l'on fait un test de Mann-Whitney pour comparer deux à deux les distributions des vraisemblances finales obtenues, toutes les p-valeurs se situent entre 0,15 et 0,96 ; aucune option ne donne donc des résultats significativement meilleurs ou moins bons. On s'attendrait pourtant à ce que lorsque l'on ne remet jamais la température à T_0, la vraisemblance obtenue soit moins bonne car l'espace des solutions est moins exploré que lorque l'on remet de temps en temps la température à T_0. Il est probable que ce ne soit pas le cas à cause de la taille réduite des données. En effet, l'espace des solutions est lui-même de taille réduite, donc un seul parcours du cooling schedule permettrait de l'explorer aussi bien que plusieurs parcours du cooling schedule.

Le choix d'option se portera donc sur une remise à T_0 de la température tous les 300 décréments de celle-ci, étant donné que cette option réduit le temps de recherche par rapport aux autres.



(a) (b)

FIG. 6.7 – (a) Evolution de la vraisemblance (moyenne sur 50 réplicats) et (b) moyenne et écart-type de la vraisemblance finale (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus avec différentes options de remise de la température à T_0. Les droites horizontales tracées en noir correspondent à la vraisemblance du vrai arbre phylogénétique. *NEVER* : Aucune remise de la température à T_0. *DECR*x : remise de la température à T_0 toutes les x diminutions de la température. *POURCENT* : remise de la température à T_0 lorsque la température atteint 0,001% de T_0.

	jamais	0,001%	apr. 100 décr.	apr. 200 décr.	apr. 300 décr.
$-\ln(L)$	14018.86	14033.52	14047.01	14005.44	14014.92
écart-type	147.70	124.30	164.75	165.34	134.12
Temps (s)	18.44	18.09	18.19	18.93	12.87
écart-type	6.01	5.79	6.68	7.60	4.25

TAB. 6.8 – Moyenne et écart-type de la vraisemblance et du temps de calcul obtenus avec différentes options de remise de la température du SA à T_0. 0,001% : remise de la température à T_0 lorsque la température atteint 0,001% de T_0. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

Calcul de ΔL

La figure 6.8 montre l'évolution de la vraisemblance en fonction du temps pour les deux options de calcul de ΔL. Sans discussion, lorsque ΔL est calculée comme valant 0,1% de la vraisemblance de l'arbre construit par NJ, les résultats sont bien meilleurs (valeur moyenne des vraisemblances finales obtenues données table 6.9). La p-valeur obtenue avec un test de Mann-Whitney, qui compare les moyennes des distributions des vraisemblances finales avec les deux options, est inférieure à 10^{-15}.

	Période « burn-in »	$0,1\%L_{NJ}$
$-\ln(L)$	15639.22	14014.92
écart-type	364.86	134.12
Temps (s)	6.56	12.87
écart-type	0.19	4.25

TAB. 6.9 – Vraisemblance et temps de calcul obtenus avec deux manières de calculer ΔL pour le SA à T_0. $0,1\%L_{NJ}$: ΔL vaut 0,1% de la vraisemblance de l'arbre construit par NJ. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

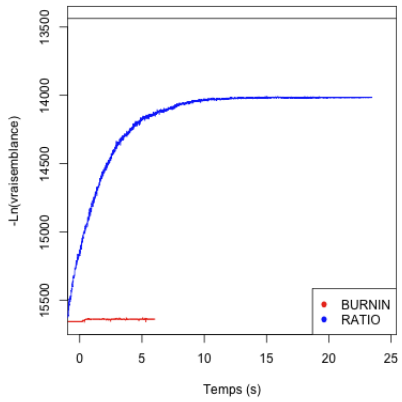


FIG. 6.8 – Evolution de la vraisemblance pour deux manières différentes de calculer ΔL pour le SA. La droite horizontale tracée en noir correspond à la vraisemblance du vrai arbre phylogénétique. *BURNIN* : calculé avec une période « burn-in ». *RATIO* : ΔL vaut 0,1% de la vraisemblance de l'arbre construit par NJ

6.1.3 Paramètres testés pour le GA

Différentes tailles de population ont été testées, ainsi que plusieurs types de sélection.

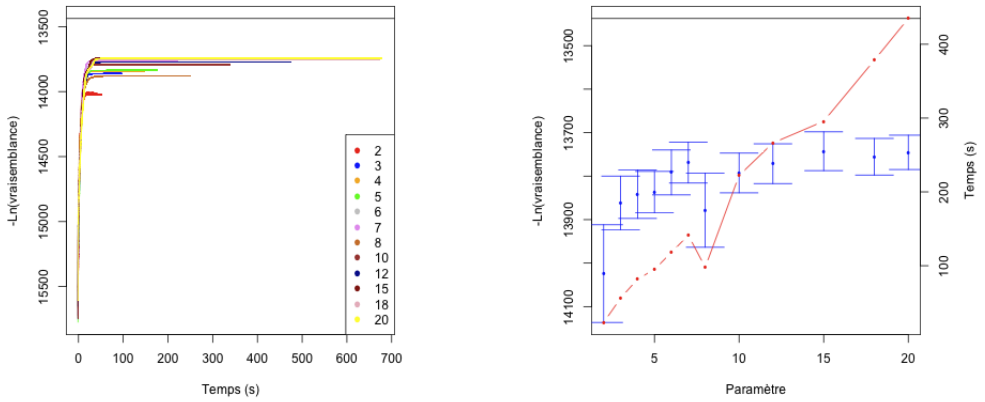
Taille de la population

L'évolution de la vraisemblance pour différentes tailles de population est montrée figure 6.9(a), et les valeurs moyennes de vraisemblances finales et de temps de calcul, figure 6.9(b). On voit que pour deux individus, la vraisemblance finale est fort en dessous de ce qu'on obtient pour des populations plus grandes. De 3 à 7 individus, la vraisemblance augmente proportionnellement au nombre d'individus, et au-delà de 7 individus, on n'observe plus une nette tendance à l'augmentation.

En ce qui concerne le temps de recherche, il augmente de façon très régulière avec la taille de la population. La seule exception est pour la population de huit individus, pour laquelle on observe une diminution de la vraisemblance par rapport aux populations de 7 individus, couplé à une diminution de temps. Il est probable que ce soit un artefact dû au nombre réduit de réplicats faits. Il suffit de très peu de réplicats donnant un arbre de vraisemblance très basse et atteignant la condition d'arrêt très tôt pour tirer les deux moyennes vers le bas.

Les valeurs des moyennes, ainsi que les écarts-type sur les vraisemblances finales et le temps de recherche sont donnés table 6.10.

La table 6.11 donne les p-valeurs de test de Mann-Whitney permettant de comparer deux à deux les moyennes des distributions des vraisemblances finales obtenues pour les différentes tailles de population. Avec une population de 2 individus, la vraisemblance moyenne est significativement inférieure à celle obtenue avec une population plus grande. La population de 6 individus donne des résultats significativement meilleurs que les populations de taille inférieure, tandis qu'à partir de 7 individus, les résultats ne sont plus significativement meilleurs, sauf pour 15 et 20 individus (mais pas pour 18, ce qui peut faire douter de la confiance à accorder à ces résultats), mais on double alors le temps de recherche. Le meilleur choix semble donc 6 individus.



(a) (b)

FIG. 6.9 – (a) Evolution de la vraisemblance en fonction du temps (moyenne sur 50 réplicats) et (b) moyenne et écart-type de la vraisemblance finale et temps de calcul moyens obtenus avec diverses tailles de population pour le GA. La droite horizontale tracée en noir correspond à la vraisemblance du vrai arbre phylogénétique.

nombre d'individus	2	3	4	5	6	7
$-\ln(L)$	14024.12	13861.59	13841.40	13836.71	13791.39	13768.81
écart-type	224.86	123.09	111.53	95.35	103.31	93.83
Temps (s)	23.07	56.10	81.81	95.00	117.82	141.42
écart-type	14.11	21.46	25.16	32.33	32.16	40.00
nombre d'individus	8	10	12	15	18	20
$-\ln(L)$	13878.26	13792.65	13771.49	13742.71	13755.49	13745.34
écart-type	170.39	91.45	91.59	89.42	84.39	79.06
Temps (s)	97.73	222.24	266.31	295.66	378.57	434.89
écart-type	66.05	52.39	74.77	71.37	106.44	121.94

TAB. 6.10 – Moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus avec diverses tailles de population pour le GA. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

Sélection appliquée

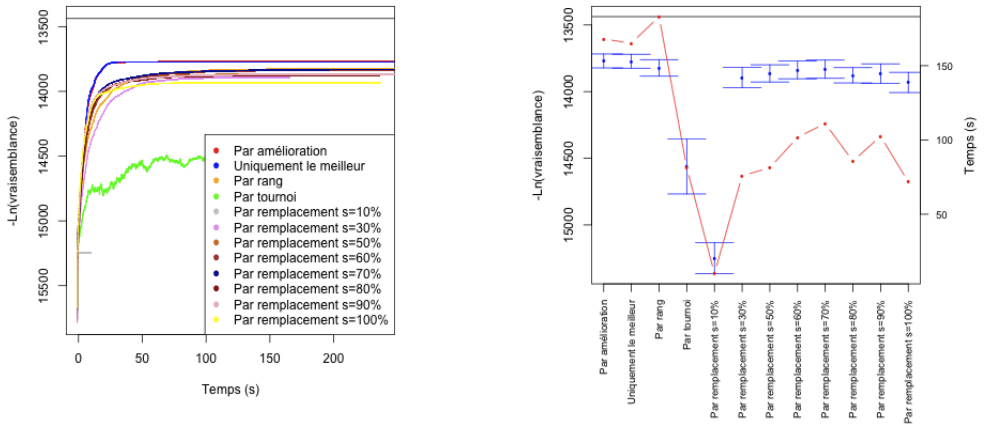
La figure 6.10(a) montre l'évolution de la vraisemblance pour les différents types de sélection testés, tandis que la figure 6.10(b) montre les moyennes et écarts-type des vraisemblances finales obtenues, ainsi que les moyennes des temps de recherche des arbres pour ces types de sélection (valeurs données table 6.12).

La sélection par tournoi, ainsi que par remplacement avec une force de sélection de 0,1 donnent visiblement de moins bons résultats que les autres sélections. Ce deux sélection permettant à des individus très mauvais d'être sélectionnés, cela n'est pas particulièrement étonnant, à ceci près qu'il semblerait qu'il suffise d'augmenter à 0,2 la force de sélection pour le remplacement, pour obtenir des résultats bien meilleurs, et qui s'améliorent plus lorsque l'on augmente encore la force de sélection.

Les deux sélection donnant les meilleurs résultats sont la sélection par amélioration et la sélection du meilleur. La table 6.13, donnant les résultats de tests de Mann-Whitney performés sur chaque paire de distributions de vraisemblances possible montre que ces deux types de sélections donnent effectivement des résultats significativement meilleurs que les autres sélections, à l'exception de

Taille pop.	3	4	5	6	7	8	10	12	15	18	20
2	2.49×10^{-5}	2.45×10^{-6}	1.15×10^{-6}	2.87×10^{-9}	4.49×10^{-11}	3.07×10^{-4}	1.12×10^{-9}	9.81×10^{-11}	1.55×10^{-12}	5.30×10^{-12}	4.20×10^{-13}
3		0.58	0.62	0.010	1.28×10^{-4}	0.51	0.0056	2.05×10^{-4}	6.56×10^{-7}	3.79×10^{-6}	1.70×10^{-7}
4			0.89	0.042	1.23×10^{-3}	0.20	0.032	0.0022	1.18×10^{-5}	1.28×10^{-4}	6.85×10^{-6}
5				0.027	5.32×10^{-4}	0.23	0.020	9.95×10^{-4}	6.22×10^{-6}	2.34×10^{-5}	2.81×10^{-6}
6					0.29	3.58×10^{-3}	0.95	0.39	0.019	0.11	0.024
7						1.56×10^{-4}	0.23	0.94	0.16	0.49	0.27
8							0.0034	1.28×10^{-4}	1.08×10^{-6}	1.18×10^{-5}	1.52×10^{-6}
10								0.23	0.0063	0.046	0.014
12									0.097	0.48	0.23
15										0.36	0.64
18											0.65

TAB. 6.11 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des $-\ln(L)$ pour différentes tailles de populations pour le GA



(a) (b)

FIG. 6.10 – (a) Evolution de $-\ln(L)$ en fonction du temps (moyenne sur 50 réplicats) et (b) moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus avec divers types de sélection pour le GA. La droite horizontale tracée en noir correspond à $-\ln(L)$ du vrai arbre phylogénétique.

	Amélioration	Meilleur	Rang	Tournoi	Rempl. s=0,1	Rempl. s=0,3
$-\ln(L)$	13769.94	13772.64	13822.37	14561.94	15248.62	13894.45
écart-type	104.65	104.51	122.85	411.44	232.81	151.03
Temps (s)	167.96	165.19	183.04	81.37	10.03	75.91
écart-type	61.80	57.58	96.61	31.17	0.34	29.94
	Rempl. s=0,5	Rempl. s=0,6	Rempl. s=0,7	Rempl. s=0,8	Rempl. s=0,9	Rempl. s=1
$-\ln(L)$	13863.67	13837.30	13831.47	13877.39	13865.16	13931.56
écart-type	129.22	133.93	136.45	115.39	145.57	151.48
Temps (s)	81.27	101.76	111.10	85.69	102.53	71.68
écart-type	59.68	54.92	68.19	52.23	70.57	42.89

TAB. 6.12 – Moyenne et écart-type de $-\ln(L)$ et du temps de calcul obtenus avec divers types de sélection pour le GA. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

la sélection par rang, pour laquelle la p-valeur vaut 0,06 et 0,07 par rapport à la sélection par amélioration et à la sélection du meilleur respectivement.

Ces trois types de sélection devraient donc être privilégiés, quoique la sélection du meilleur uniformise les individus de la population à chaque itération de l'algorithme, ce qui risque de bloquer celui-ci dans un optimum local. Cela ne prête pas à conséquence sur un set de données de cette taille, mais il est sans doute bon de garder un peu de variété dans la population dès lors que l'on travaille sur des sets de données plus grand.

	Meilleur	Rang	Tournoi	Rempl. s=0,1	Rempl. s=0,3	Rempl. s=0,5	Rempl. s=0,6	Rempl. s=0,7	Rempl. s=0,8	Rempl. s=0,9	Rempl. s=1
Amélioration	0.95	0.061	1.27×10^{-17}	7.07×10^{-18}	2.34×10^{-5}	1.11×10^{-4}	0.019	0.032	4.64×10^{-6}	1.15×10^{-3}	5.46×10^{-8}
Meilleur		0.069	1.45×10^{-17}	7.07×10^{-18}	2.73×10^{-5}	3.16×10^{-4}	0.016	0.044	1.78×10^{-5}	1.57×10^{-3}	1.70×10^{-7}
Rang			4.73×10^{-17}	7.067×10^{-18}	8.71×10^{-3}	0.036	0.54	0.86	4.76×10^{-3}	0.14	8.39×10^{-5}
Tournoi				2.65×10^{-13}	8.88×10^{-16}	2.22×10^{-16}	$< 10^{-15}$	$< 10^{-15}$	2.22×10^{-16}	4.44×10^{-16}	4.44×10^{-15}
Rempl. s=0,1					$< 10^{-15}$	$< 10^{-15}$	$< 10^{-15}$	$< 10^{-15}$	$< 10^{-15}$	$< 10^{-15}$	$< 10^{-15}$
Rempl. s=0,3						0.40	0.044	0.030	0.67	0.30	0.23
Rempl. s=0,5							0.22	0.12	0.65	0.80	0.027
Rempl. s=0,6								0.76	0.058	0.37	1.33×10^{-3}
Rempl. s=0,7									0.024	0.26	6.86×10^{-4}
Rempl. s=0,8										0.37	0.064
Rempl. s=0,9											0.022

TAB. 6.13 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des $-\ln(L)$ pour différents types de sélection pour le GA

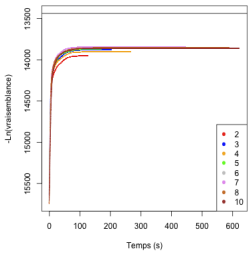
6.1.4 Paramètres testés sur le CP

Le nombre de populations évoluant en parallèle, ainsi que la taille de celles-ci, le type de consensus, la sélection des opérateurs par rapport au consensus, et enfin le niveau de tolérance ont été testés pour le CP.

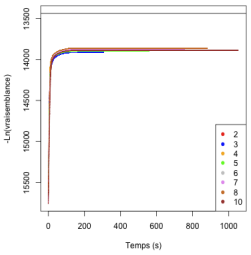
Nombre et taille de population

La figure 6.11 montre l'évolution de la vraisemblance lors d'un CP avec 2, 3, 4 et 5 populations, pour diverses tailles de populations. Les moyennes des vraisemblances finales et des temps de recherche sont montrés figure 6.12 (valeurs données table 6.14).

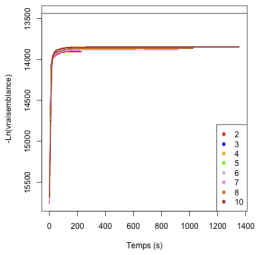
Si l'on voit que le temps de recherche augmente proportionnellement au nombre d'individus dans les populations (comme pour le GA), il est difficile d'après les graphiques, et les valeurs, de tirer une conclusion quant au nombre d'individus approprié à chaque taille de métapopulation. Un test de Mann-Whitney a donc été fait pour chaque paire de distributions possible des vraisemblances finales obtenues avec 2, 3, 4 et 5 populations. Les p-valeurs de ces tests sont données table 6.15. Presqu'aucune des moyennes n'est significativement différente de celles obtenues avec le même nombre et une taille différente de population.



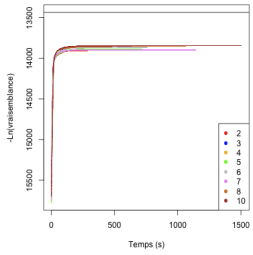
(a) 2 populations



(b) 3 populations



(c) 4 populations

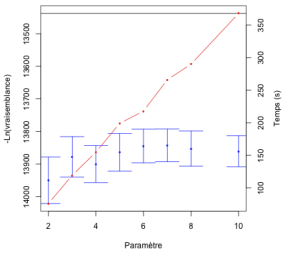


(d) 5 populations

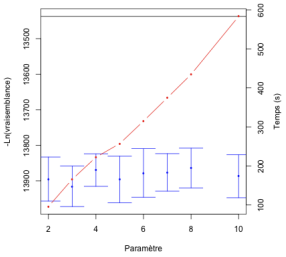
FIG. 6.11 – Evolution de $-\ln(L)$ pour différents nombres et différentes tailles de populations pour le CP (moyenne sur 50 réplicats). (a) 2 populations, (b) 3 populations, (c) 4 populations et (d) 5 populations. La droite horizontale tracée en noir correspond à $-\ln(L)$ du vrai arbre phylogénétique.

L'influence du nombre de populations a été testée en comparant les résultats obtenus pour 2, 3, 4 et 5 populations de 4 individus chacune. Les résultats sont montrés figure 6.13 (valeurs : voir table 6.14). A nouveau, des tests de Mann-Whitney ont été faits, et les p-valeurs sont données table 6.16. Seule la différence entre la moyenne pour 2 populations et 4 populations est inférieur à 0,05. Comme pour le nombre d'individus, il est impossible de tirer une conclusion quant à la taille de la métapopulation donnant le meilleur résultat.

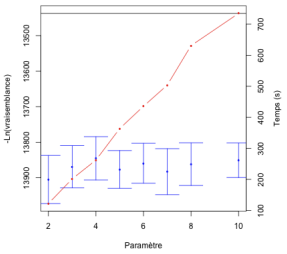
Ces tests n'ont donc pas permis de tirer une conclusion quant à la meilleure combinaison (nombre de populations - taille des populations). Il est fort possible que sur des données plus grandes, des différences apparaissent plus clairement.



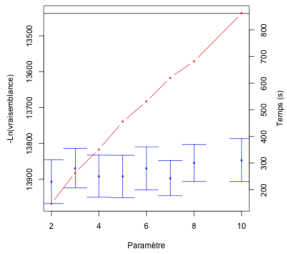
(a) 2 populations



(b) 3 populations



(c) 4 populations



(d) 5 populations

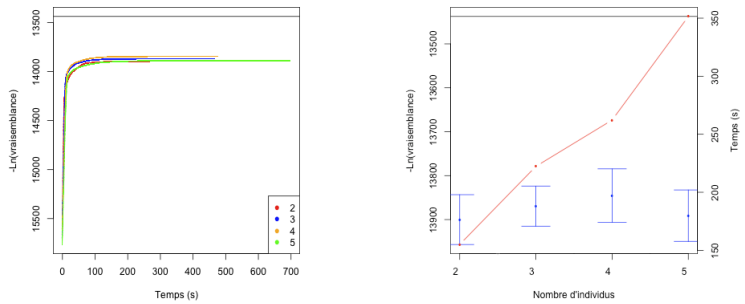
FIG. 6.12 – Moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche), et moyenne du temps de recherche moyen (en rouge - axe de droite) obtenus avec différents nombres et différentes tailles de populations pour le CP. (a) 2 populations, (b) 3 populations, (c) 4 populations et (d) 5 populations.

2 pop.	2 ind.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
$-\ln(L)$	13950.00	13878.03	13900.13	13864.15	13844.64	13842.81	13852.54	13860.67
écart-type	143.11	124.08	113.60	115.55	103.18	100.47	107.72	95.05
Temps (s)	75.98	119.39	154.92	199.07	217.95	266.05	289.81	367.77
écart-type	25.20	29.75	48.00	57.51	53.25	59.30	98.89	92.69
3 pop.	2 ind.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
$-\ln(L)$	13894.70	13915.08	13869.51	13895.90	13877.47	13875.95	13863.70	13886.71
écart-type	123.85	114.10	91.34	130.94	136.89	105.43	111.92	120.83
Temps (s)	95.78	165.63	222.30	257.22	314.93	374.52	434.50	583.28
écart-type	29.79	53.88	67.86	72.15	91.53	107.83	118.01	156.76
4 pop.	2 ind.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
$-\ln(L)$	13905.11	13869.10	13845.47	13876.62	13859.36	13882.90	13861.85	13850.83
écart-type	135.49	118.91	121.95	106.27	112.34	129.36	120.15	97.47
Temps (s)	121.96	201.53	262.22	363.81	435.69	503.78	630.95	734.78
écart-type	35.82	63.65	82.34	93.28	120.26	150.20	161.64	184.71
5 pop.	2 ind.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
$-\ln(L)$	13906.82	13868.88	13891.24	13892.46	13869.70	13896.61	13854.65	13846.48
écart-type	122.05	110.41	117.21	118.30	120.08	98.35	103.65	120.70
Temps (s)	147.83	264.34	351.53	455.18	530.97	620.57	681.56	861.85
écart-type	47.17	77.01	102.77	128.06	129.01	163.97	183.14	244.40

TAB. 6.14 – Moyenne et écart-type de $-\ln(L)$ et du temps de calcul obtenus avec différents nombres et différentes tailles de populations pour le CP. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

2 pop.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
2 ind.	0.010	0.13	2.04×10^{-3}	1.32×10^{-4}	5.59×10^{-5}	4.44×10^{-4}	4.22×10^{-4}
3 ind.		0.30	0.75	0.21	0.21	0.42	0.64
4 ind.			0.093	0.0096	0.0065	0.046	0.044
5 ind.				0.32	0.32	0.57	0.85
6 ind.					0.88	0.64	0.38
7 ind.						0.62	0.33
8 ind.							0.67
3 pop.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
2 ind.	0.22	0.53	0.77	0.49	0.70	0.38	0.96
3 ind.		0.049	0.58	0.11	0.097	0.046	0.31
4 ind.			0.18	0.93	0.77	0.79	0.45
5 ind.				0.27	0.36	0.15	0.57
6 ind.					0.84	0.76	0.59
7 ind.						0.67	0.62
8 ind.							0.36
4 pop.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
2 ind.	0.33	0.057	0.47	0.16	0.77	0.099	0.068
3 ind.		0.30	0.83	0.52	0.63	0.61	0.42
4 ind.			0.15	0.65	0.15	0.69	0.79
5 ind.				0.34	0.81	0.31	0.22
6 ind.					0.19	0.87	0.67
7 ind.						0.24	0.099
8 ind.							0.98
5 pop.	3 ind.	4 ind.	5 ind.	6 ind.	7 ind.	8 ind.	10ind.
2 ind.	0.091	0.49	0.41	0.11	0.56	0.015	0.014
3 ind.		0.31	0.36	0.89	0.15	0.37	0.25
4 ind.			0.91	0.37	0.84	0.087	0.051
5 ind.				0.33	0.74	0.092	0.058
6 ind.					0.17	0.41	0.34
7 ind.						0.027	0.028
8 ind.							0.66

TAB. 6.15 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des $-\ln(L)$ obtenus avec différents nombres et différentes tailles de populations pour le CP



(a) (b)

FIG. 6.13 – (a) Evolution de $-\ln(L)$ en fonction du temps (moyenne sur 50 réplicats) et (b) moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus avec différents nombres de populations de 4 individus pour le CP. La droite horizontale tracée en noir correspond à $-\ln(L)$ du vrai arbre phylogénétique.

	3 pop.	4 pop.	5 pop.
2 pop.	0.10	0.024	0.72
3 pop.		0.30	0.32
4 pop.			0.066

TAB. 6.16 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions de $-\ln(L)$ obtenus avec différents nombres de populations de 4 individus pour le CP

Type de consensus

L'évolution de la vraisemblance avec les consensus de type strict et stochastique est montrée figure 6.14. Ce graphique ne montre pas de différence évidente, le consensus stochastique semble simplement atteindre un plateau de vraisemblance un peu plus vite que le consensus strict. Les moyennes de vraisemblances finales obtenues avec ces deux consensus sont en effet fort proches (table 6.17). Ceci est confirmé par un test de Mann-Whitney (p-valeur = 0,63).

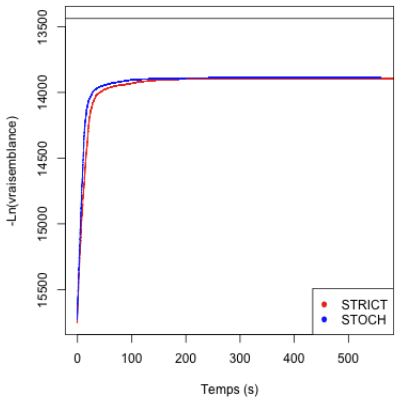


FIG. 6.14 – Evolution de la vraisemblance en fonction du temps (moyenne sur 50 réplicats) obtenue avec les deux types de consensus du CP. La droite horizontale tracée en noir correspond à la vraisemblance du vrai arbre phylogénétique.

	Cons. strict	Cons. stoch.
$-\ln(L)$	13891.93	13885.96
écart-type	110.60	96.88
Temps (s)	360.05	322.37
écart-type	98.81	93.33

TAB. 6.17 – Vraisemblance et temps de calcul moyens (avec écarts-type) obtenus avec les deux types de consensus du CP. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

Sélection des opérateurs par rapport branches consensus

Lorsque la sélection des opérateurs par rapport aux branches consensus est supervisée, les résultats sont visiblement meilleurs que lorsque la sélection est aveugle (figure 6.15 et table 6.18). Ceci est confirmé par un test de Mann-Whitney (p-valeur = 0,015), montrant que la moyenne des vraisemblances finales pour la sélection supervisée est significativement supérieure à la moyenne des vraisemblances finales pour la sélection aveugle. Le temps de recherche semble également plus court avec la sélection supervisée. Il est cependant possible qu'avec un set de données plus important, la sélection supervisée ralentisse l'algorithme. En effet, ici, il est fort probable que de nombreuses branches consensus apparaissent très tôt dans l'algorithme, vu l'espace des solutions réduit, et il vaut donc la peine de superviser la sélection des opérateurs dès le début. Pour des sets de données plus grand par contre, des branches consensus apparaîtront plus tard dans l'algorithme, et de nombreux calculs seront fait dès le début pour savoir quels opérateurs peuvent encore être appliqués aux arbres alors que la probabilité de choisir par hasard un opérateur cassant une branche consensus restera très faible (voir nulle) pendant longtemps.

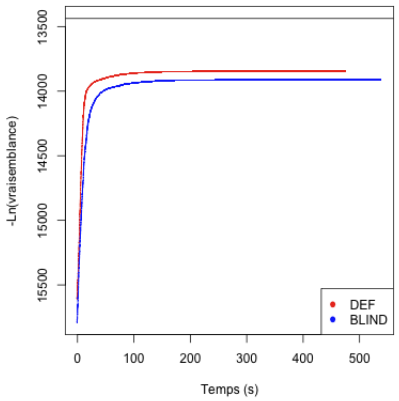


FIG. 6.15 – (a) Evolution de la vraisemblance en fonction du temps (moyenne sur 50 réplicats) et (b) vraisemblance et temps de calcul moyens obtenus avec les deux types de sélection des opérateurs par rapport aux branches consensus du CP. La droite horizontale tracée en noir correspond à la vraisemblance du vrai arbre phylogénétique. *DEF* : sélection par défaut, c'est-à-dire supervisée. *BLIND* : sélection aveugle des opérateurs.

	Cons. strict	Cons. stoch.
$-\ln(L)$	13845.47	13912.60
écart-type	121.95	134.40
Temps (s)	262.22	290.08
écart-type	82.34	80.14

TAB. 6.18 – Vraisemblance et temps de calcul moyens (avec écarts-type) obtenus avec les deux types de sélection des opérateurs par rapport aux branches consensus du CP. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

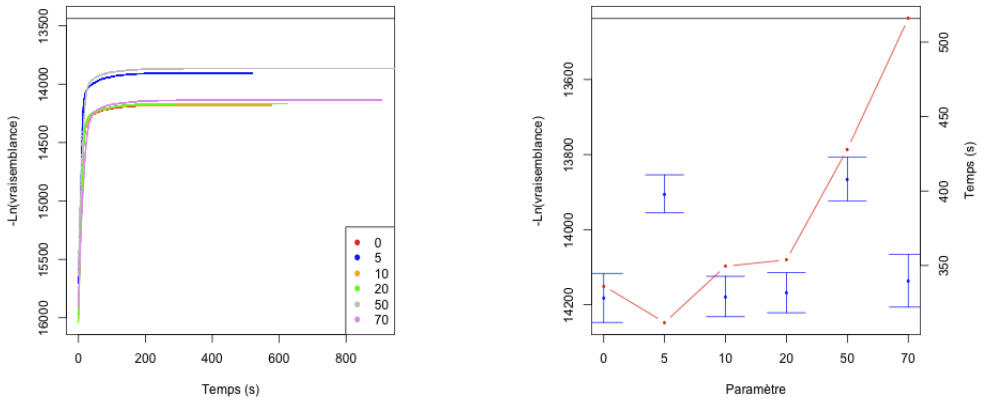
Tolérance

Les niveaux de tolérance de 5% et 50% se détachent nettement des autres niveaux en terme de vraisemblance finale (figure 6.16 et table 6.19, confirmé par des test de Mann-Whitney, table 6.20). C'est assez étonnant. On retrouve vraiment deux « groupes » de vraisemblances finales, et les niveaux de tolérance dans chacun de ces groupe n'est pas vraiment cohérent. On se serait plutôt attendu à de bons résultat pour une basse tolérance, et une dégradation de ces résultats au fur et à mesure que le niveau de tolérance monte. Et les écarts-type sont du même ordre pour tous les niveaux de tolérance ; par conséquent, il ne s'agit sans doute pas d'un artefact dû au manque de données. Il serait donc intéressant de mener d'autres test afin de comprendre ces résultats.

Tolérance	0%	5%	10%	20%	50%	70%
$-\ln L$	14181.99	13904.48	14177.95	14167.92	13864.91	14136.08
écart-type	130.57	100.92	107.32	106.92	116.70	139.97
Temps (s)	336.30	311.70	349.49	353.73	428.15	515.89
écart-type	88.32	97.69	106.63	107.26	152.71	160.25

TAB. 6.19 – Vraisemblance et temps de calcul moyens (avec écarts-type) obtenus avec différents niveaux de tolérance pour le CP. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

Dans l'immédiat, la tolérance à 5% semble la plus indiquée, car les résultats sont aussi bons que pour 50% mais le temps de recherche semble plus court.



(a) (b)

FIG. 6.16 – (a) Evolution de la vraisemblance en fonction du temps (moyenne sur 50 réplicats) et (b) vraisemblance et temps de calcul moyens obtenus avec différents niveaux de tolérance pour la CP. La droite horizontale tracée en noir correspond à la vraisemblance du vrai arbre phylogénétique.

Tolérance	5%	10%	20%	50%	70%
0%	6.22×10^{-15}	0.83	0.45	1.11×10^{-15}	0.082
5 %		5.97×10^{-16}	4.50×10^{-16}	0.10	8.97×10^{-13}
10%			0.47	$< 10^{-15}$	0.069
20%				2.22×10^{-16}	0.21
50%					1.92×10^{-14}

TAB. 6.20 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des vraisemblances obtenus avec différents niveaux de tolérance pour le CP

6.2 Comparaison des heuristiques

Les quatre heuristiques ont été paramétrées d'après les résultats des tests ci-dessus, puis testées à la fois sur le même set de données simulé que précédemment, ainsi que sur le set de données réelles, dont la taille (111 taxa $\times$ 3679 caractères) correspond plus aux sets de données pour lesquels MetaPIGA a été créé.

Les paramètres généraux utilisés sont les suivants : arbre(s) de départ construit(s) par Loose Neighbour-Joining (10%), opérateur BLM désactivé et opérateurs sélectionnés selon des fréquences dynamiques.

Pour le SA, le cooling schedule choisi est Lundy, avec décrément toutes les 10 améliorations ou 100 diminutions de la vraisemblance et remise à T_0 après 300 décréments. ΔL vaut 0,1% de la vraisemblance de l'arbre construit par NJ.

Le GA travaille avec une population de 6 individus, à laquelle est appliquée une sélection par amélioration.

Quant au CP, il utilise 4 populations de 4 individus, une sélection par amélioration et un consensus stochastique. La sélection des opérateurs est supervisée.

Tous les autres paramètres sont les mêmes que les paramètres par défaut utilisés pour les tests et décrits section 5.2.1.

6.2.1 Données simulées

La figure 6.17 montre les résultats obtenus pour la comparaison des heuristiques avec les données simulées, et les valeurs moyennes des vraisemblances finales ainsi que les temps de recherche sont donnés table 6.21.

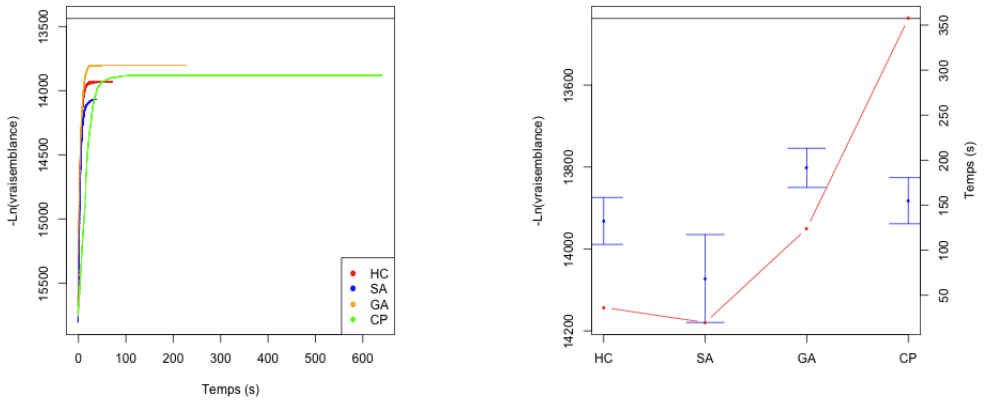
Le GA trouve les arbres de plus haute vraisemblance, tandis que le SA donnent visiblement les moins bons résultats (tests de Mann-Whitney effectués pour comparer les moyennes, résultats table 6.22). Le SA semble de plus s'arrêter très tôt, et bien avant d'avoir atteint un plateau, comme on peut en observer un pour les trois autres heuristiques. Peut-être la condition d'arrêt utilisée n'est-elle pas adaptée pour cette heuristique en particulier. Le CP est moins bon que le GA, mais meilleur que le HC, quoique cela soit à peine significatif.

Le CP semble donc à la fois être plus lent et donner de moins bons résultats que le GA , ce qui est en contradiction avec les résultats présentés lors de la sortie de la première version de MetaPIGA (Lemmon and Milinkovitch, 2002). Cependant, le test présenté dans cet article a été fait sur des données comprenant 80 taxa, et non 20, et d'autres tests montrent que le metaGA ne devient plus efficace que les autres heuristiques que pour de grands sets de données.

Les très bons résultats obtenus par HC sont aussi à imputer à la taille réduite des données. L'espace des solutions étant très réduit, il y a évidemment moins de pics de vraisemblances ne correspondant pas à l'optimum global dans lesquels l'algorithme pourrait rester bloqué.

Le SA, qui est supposé être au moins aussi bon que le HC, donne de très mauvais résultats. Soit la condition d'arrêt est atteinte trop tôt, soit trop de diminutions de la vraisemblance sont acceptées par l'algorithme. Quoi qu'il en soit, le paramètrage de cette heuristique nécessite visiblement des tests supplémentaires.

Quant au GA, sa faculté à éviter les optimum locaux semble lui permettre d'inférer des arbres de plus haute vraisemblance que les autres heuristiques. Cela se fait cependant au détriment de la rapidité : le HC étant plus de 3 fois plus rapide que le GA.



(a) (b)

FIG. 6.17 – (a) Evolution de $-\ln(L)$ en fonction du temps (moyenne sur 50 réplicats) et (b) moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus pour les quatre heuristiques avec les données simulées. La droite horizontale tracée en noir correspond à $-\ln(L)$ du vrai arbre phylogénétique.

Heuristique	HC	SA	GA	CP
$-\ln(L)$	13931.63	14072.12	13802.07	13882.08
écart-type	114.52	214.60	95.43	112.62
Temps (s)	36.40	19.75	124.04	357.32
écart-type	9.80	7.66	37.61	96.34

TAB. 6.21 – Moyenne et écart-type de $-\ln(L)$ et du temps de calcul obtenus avec les différentes heuristiques pour le set de données simulé. ($-\ln(L)$ du vrai arbre phylogénétique = 13437,5235.)

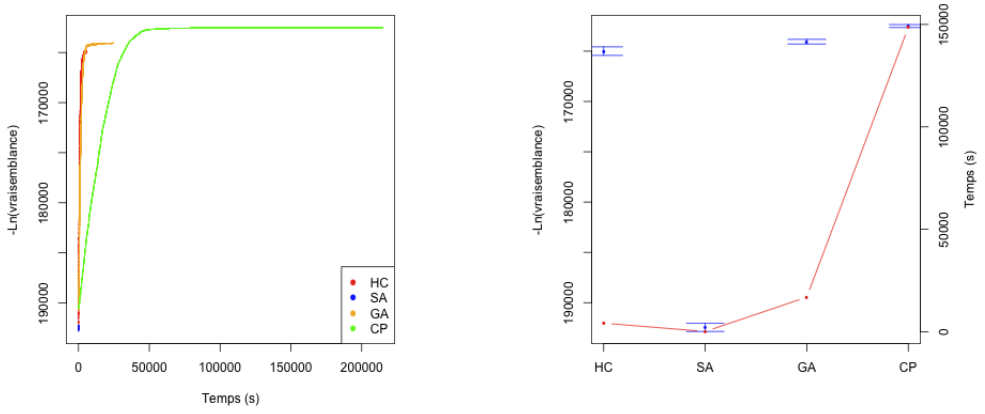
Heuristique	SA	GA	CP
HC	3.42×10^{-4}	3.85×10^{-8}	0.025
SA		7.80×10^{-12}	1.24×10^{-6}
GA			7.03×10^{-4}

TAB. 6.22 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des $-\ln(L)$ obtenus avec les différentes heuristiques pour le set de données simulé.

6.2.2 Données réelles

La figure 6.18 montre les résultats obtenus pour la comparaison des heuristiques appliquées aux données réelles (moyennes des vraisemblances finales et des temps de calculs donnés table 6.23). Le CP donne de bien meilleurs résultats que sur les données simulées ; il semble en effet meilleur que le HC et le GA, lesquels donnent à peu près les mêmes résultats.

Quant au SA, il atteint très rapidement la condition d'arrêt, à tel point qu'on ne le voit pas sur la figure 6.18(a), montrant l'évolution de la vraisemblance en fonction du temps pour les quatre heuristiques. Cela confirme les résultats obtenus avec les données simulées : avec les paramètres actuels, le SA ne fonctionne pas bien du tout.



(a) (b)

FIG. 6.18 – (a) Evolution de $-\ln(L)$ en fonction du temps (moyenne sur 50 réplicats) et (b) moyenne et écart-type de $-\ln(L)$ (en bleu - axe de gauche) et moyenne du temps de calcul (en rouge - axe de droite) obtenus pour les quatre heuristiques avec les données réelles.

Les tests de Mann-Whitney montrent que les résultats du CP sont bien significativement supérieurs aux résultats des autres heuristiques, mais aussi que les résultats du GA sont significativement supérieurs à ceux du HC. On s'attendait cependant à ce que le HC donne de beaucoup moins bon résultats, a fortiori après avoir constaté que même sur un petit set de données, le GA donnait de

Heuristique	HC	SA	GA	CP
$-\ln(L)$	164993.86	192450.24	164056.77	162501.62
écart-type	850.52	828.43	465.76	289.65
Temps (s)	4174.31	113.51	16879.71	148509.36
écart-type	738.63	29.05	4724.06	39786.38

TAB. 6.23 – Moyenne et écart-type de $-\ln(L)$ et du temps de calcul obtenus avec les différentes heuristiques pour le set de données réelles.

meilleurs résultats que le HC.

Un autre résultat étonnant est l'évolution de la vraisemblance pour le GA et le CP. D'après les résultats présentés dans (Lemmon and Milinkovitch, 2002) (figure 6.19), le CP devrait atteindre un plateau plus rapidement que le GA. Or ici, on voit que la vraisemblance monte très lentement pour le CP, et beaucoup plus vite pour le GA. Il est possible que la cause de ces résultats étonnants soit le manque de données, puisque seuls 10 réplicats ont pu être faits sur les données réelles.

Heuristique	SA	GA	CP
HC	1.08×10^{-5}	2.88×10^{-3}	1.08×10^{-5}
SA		1.08×10^{-5}	1.08×10^{-5}
GA			1.08×10^{-5}

TAB. 6.24 – P-valeurs d'un test de Mann-Whitney permettant de comparer les moyennes des distributions des $-\ln(L)$ obtenus avec les différentes heuristiques pour le set de données réelles.

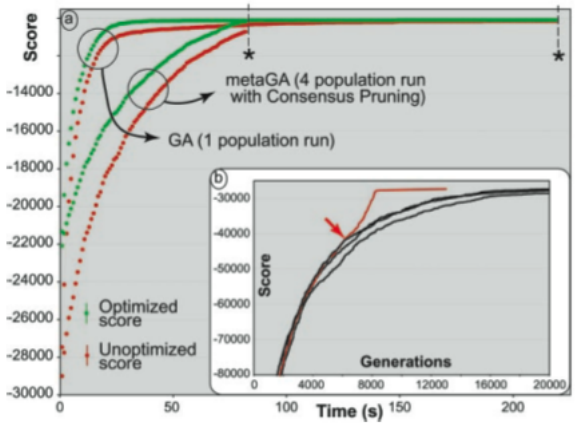


FIG. 6.19 – (Lemmon and Milinkovitch, 2002) (a) scores ($\ln(L)$) optimisés et non optimisés en fonction du temps pour le GA et le CP (80 taxa). L'astérisque indique à quel moment la condition d'arrêt a été atteinte (condition d'arrêt : tous les opérateurs applicables sans casser le consensus ont été tentés et aucun n'améliore la vraisemblance pour le meilleur arbre de chaque population). (b) Evolution de $\ln(L)$ en fonction du temps pour quatre population évoluant en parallèle sans interagir (320 taxa). La flèche rouge indique le moment à partir duquel l'une des populations a pu utiliser l'informations venant des autres poulations pour utiliser le CP, et la courbe rouge représente l'évolution de $\ln(L)$ pour cette population particulière.

Chapitre 7

Conclusion et perspectives

7.1 Conclusion

A partir des tests effectués sur certains paramètres des différentes heuristiques, on peut tirer quelques conclusions, présentées ci-après.

Il est important de garder à l'esprit que ces conclusions ne sont valables que pour un set de données relativement réduit, alors que MetaPIGA a été créé pour répondre au besoin de programmes d'inférence phylogénétique capables de traiter des données volumineuses.

En ce qui concerne le choix des arbres de départ des algorithmes, commencer avec l'arbre construit par NJ semble donner les meilleurs résultats, pour les quatres heuristiques.

La longueur des branches terminales a visiblement une influence importante sur la vraisemblance des arbres, mais est aussi sujette au phénomène d'over-fitting, pouvant éventuellement amener à être bloqué dans un optimum local qui soit loin de l'optimum global.

Quant à la sélection des opérateurs selon des fréquences dynamiques, cela ne semble pas donner de meilleurs résultat que la sélection des opérateurs au hasard.

Pour les paramètres du recuit simulé, le cooling schedule *Lundy* donne les meilleurs vraisemblances finales parmi les huit cooling schedules testés. En ce qui concerne la diminution de la température et sa remise à T_0, aucune option ne donne de résultat significativement supérieurs aux autres, mais une diminution toutes les 10 améliorations ou 100 diminutions de la température et une remise à T_0 après 300 décréments semblent réduire le temps de recherche de l'arbre. Quant au calcul de ΔL, avec la période de « burn-in », on a de bien moins bons résultats qu'en prenant $\Delta L = 0,1\%$ de la vraisemblance de l'arbre construit par NJ.

Pour les paramètres testés avec l'algorithme génétique, de bons résultats peuvent être obtenus avec 6 individus dans la population. Il est possible d'en avoir de meilleurs mais le temps de recherche est alors considérablement allongé, sans réelle certitude de faire mieux. Pour le type de sélection, les sélections par amélioration et par rang semblent les plus indiquées. La sélection du meilleur donne d'aussi bons résultats mais uniformise la population et pourrait donc avoir tendance à bloquer l'algorithme dans un optimum local, a fortiori pour de grand sets de données.

Enfin les tests effectués sur le nombre et la taille des populations d'un consensus pruning, ainsi que sur le type de consensus utilisé n'a pas permis de tirer de conclusions fermes quant aux meilleurs paramètres. Il semble par contre que la sélection supervisée des opérateurs donne de meilleurs résultats que la sélection aveugle, et qu'un niveau de tolérance de 5% (ou 50%, mais le temps de recherche est alors plus long) soit le plus efficace.

La comparaison des différentes heuristiques n'est pas particulièrement concluante.

Sur les données simulées, le GA donne les meilleurs résultats, suivi par le CP (plus de deux fois plus lent que le GA) puis le HC (trois fois plus rapide). Par contre, pour les données réelles, c'est le CP qui, comme attendu, donne les meilleurs résultats. Il est cependant très lent par rapport aux

autres heuristiques. Par ailleurs, le HC donne encore de très bons résultats alors que l'on s'attendait à ce que, sur des données aussi volumineuses, il soit bloqué dans un optimum local bien en-deçà de celui atteint par le GA ou le CP. L'algorithme de SA, quant à lui, donne de très mauvais résultats et semble atteindre très rapidement la condition d'arrêt à la fois pour les données simulées et les données réelles.

7.2 Discussion

Une première remarque générale est qu'étant donné le temps imparti pour réaliser ce travail, et le temps de calcul nécessaire aux algorithmes, seuls 50 réplicats ont pu être faits pour chaque paramètre testé. Les résultats obtenus ne sont par conséquent pas très fiables.

Ensuite, l'utilisation d'un seul set de données, de surcroît assez réduit, rend ces résultats presque inexploitables dès qu'on sort un tant soit peu du cadre fixé par ce set de données. Si l'on examine certains paramètres testés, et les résultats qu'ils ont produits, on peut montrer qu'un risque important existe pour qu'avec un set de données plus grands, les résultats soient assez différents.

Pour commencer, examinons le choix des arbres de départ. Les résultats semblent indiquer qu'il est préférable d'initialiser les algorithmes avec un arbre construit par Neighbour-Joining. L'algorithme de NJ donne de très bons résultats pour les petits sets de données, ce qui explique les résultats observés. Mais cet algorithme est moins performant pour l'inférence de grandes phylogénies. Il est donc fort possible que pour des sets de données plus grands, le NJ bloque l'algorithme dans un optimum local qui ne soit pas aussi proche du vrai arbre que pour ce set de données-ci. Par conséquent, il serait sans doute mieux, pour des données plus volumineuses, d'introduire du hasard dans la construction des arbres de départ, et donc d'utiliser plutôt le Loose Neighbour-Joining.

Ensuite, concernant la sélection des opérateurs. Une sélection au hasard semble tout aussi bonne qu'une sélection selon des fréquences dynamiques. Mais il est fort possible que ce soit la taille des données qui donne ces résultat. Avec un set de données plus grand, les opérateurs auraient plus de temps pour ajuster leur fréquence en fonction de leur efficacité, et on pourrait donc voir une amélioration des résultats avec les fréquences dynamiques, ou simplement une accélération du temps de recherche par rapport à la sélection des opérateurs au hasard.

Pour le recuit simulé, la remise régulière de la température à sa valeur initiale ne semble pas donner de meilleurs résultats que lorsque la température ne fait que décroître. Mais si on augmente la taille des données, l'espace des solutions grandit également, et de manière exponentielle. En remettant la température à sa valeur initiale, on permet à l'algorithme de repartir explorer l'espace des solutions après avoir exploré localement le pic sur lequel il était. Il paraît donc évident que cette possibilité pourrait prendre de l'importance avec un espace des solutions plus grand.

Le même argument est valable concernant la sélection appliquée à la population pour l'algorithme génétique. La sélection du meilleur semble donner de très bons résultats, mais l'uniformisation, à chaque itération, de la population risque de bloquer l'algorithme dans un optimum local. Lequel optimum local a une probabilité plus élevée d'être également optimum global dans un espace des solution pour une vingtaine de taxa que pour une centaine de taxa.

Enfin, le consensus pruning a été créé pour être appliqué à de grands sets de données. Son intérêt étant d'une part que la sélection appliquée aux individus dans chaque population peut être forte, car la collaboration des populations empêche l'algorithme de tomber dans un optimum local, d'autre part que la création de branches consensus incassables semble pouvoir accélérer l'algorithme, en particulier pour de grandes phylogénies. Cela pourrait expliquer pourquoi on n'a pu tirer presque aucune conclusion quant aux paramètres testés sur un petit set de données.

Concernant les comparaisons des heuristiques, deux remarques importantes sont à retenir. Tout d'abord, les tests de paramétrage ayant été faits sur le set de donnée simulées, les paramètres utilisés sont sans doute plus adaptés à des données de petites tailles, malgré que l'arbre de départ choisi ait été le LNJ pour éviter de tomber trop vite dans l'optimum local proche de l'arbre NJ, et que les opérateurs ait été sélectionnés selon des fréquences dynamiques (les algorithmes prenant plus

de temps sur les données réelles et laissant donc aussi plus de temps aux fréquences de s'ajuster). Ensuite, le petit nombre de réplicats faits, en particulier pour les données simulées, rend les résultats assez peu fiables.

Néanmoins, force est de reconnaître que les résultats obtenus ne ressemblent pas à ce que l'on attendait. Le CP semble beaucoup trop lent avec les données réelles, et le HC donne de meilleurs résultats qu'attendu, à moins que ce ne soit le GA et le CP qui ne donnent des résultats moins bons qu'attendu, difficile de trancher.

7.3 Perspectives

Etant données sa portée réduite, ce travail doit être considéré comme une étude préliminaires des fonctionnalités et performances de MetaPIGA. Ici sont présentées quelques pistes à explorer pour de futures recherches sur les performances de ce programme.

Pour commencer, tous les paramètres (incluant ceux testés ici, ceux qui n'ont pas été testés lors de ce travail, et ceux qui ne sont pas encore implémentés) devraient être testés sur différents sets de données : des données réelles et des données simulées ; de moyennes, grandes, et très grandes phylogénies ; ainsi que des phylogénies présentant des particularités dont on sait qu'elles rendent l'inférence phylogénétique particulièrement facile ou difficile. Une fois le paramétrage fini, de nouvelles comparaisons des heuristiques pourront être réalisées.

Ensuite, il me semble important d'examiner le rapport qu'il pourrait y avoir entre les valeurs de support des branches de l'arbre consensus construit à partir de plusieurs réplicats, et les probabilités postérieures données par les méthodes Bayesiennes, ainsi que les valeurs de support générées par bootstrap. Et par conséquent, il est essentiel de vérifier si les valeurs de support sont biaisées de la même manière que semblent l'être les probabilités postérieures, notamment en appliquant à MetaPIGA les mêmes tests que ceux fait sur MrBayes (voir section 4.3.1).

Enfin, il sera intéressant de comparer les performances de MetaPIGA à celles réalisées par d'autres programmes d'inférence phylogénétiques, en particulier MrBayes, qui est actuellement le plus utilisé, et GARLI, qui utilise aussi un algorithme génétique.

Liste des abbréviations utilisées

ADN	Acide désoxyribonucléique
ARN	Acide ribonucléique
JC	Jukes-Cantor
HKY85	Hasegawa-Kishino-Yano (1985)
TN93	Tamura-Nei (1993)
GTR	General Time Reversible
K2P	Kimura-2-paramètres
NJ	Neighbour-Joining
HC	Hill-Climbing
UPGMA	Unweight Pair Group Method with Arithmetic mean
SA	Recuit Simulé
GA	Algorithme Génétique
CP	Consensus Pruning
MCMC	Chaîne de Markov Monte Carlo
MH	Metropolis-Hastings
BLM	Mutation de la longueur des branches
BLMINT	Mutation de la longueur des branches internes
TXS	Echange de taxa
STS	Echange de sous-arbres
SPR	Détachement et rattachement d'un sous-arbre
TBR	Division et reconnection d'un arbre
NNI	Echange des plus proches voisins
RPM	Mutation des paramètres de taux de substitution
GDM	Mutation du paramètre de la distribution gamma
PIM	Mutation de la proportion d'invariants
LNJ	Loose Neighbour-Joining
MetaGA	algorithme génétique métapopulationnel
B&B	Branch and Bound
(MC^3)	Metropolis-coupled MCMC

Bibliographie

Adachi, J. and Hasegawa, M. (1995). Improved dating of the human chimpanzee separation in the mitochondrial-dna tree - heterogeneity among amino-acid sites. *Journal of Molecular Evolution*, **40**(6), 622–628.

Bossuyt, F., Brown, R., Hillis, D., Cannatella, D., and Milinkovitch, M. (2006). Phylogeny and biogeography of a cosmopolitan frog radiation : Late cretaceous diversification resulted in continent-scale endemism in the family ranidae. *Systematic Biology*, **55**(4), 579–594.

Brooks, S. P. and Morgan, B. J. T. (1995). Optimization using simulated annealing. *The Statistician*.

Catanzaro, D., Pesenti, R., and Milinkovitch, M. C. (2007). An ant colony optimization algorithm for phylogenetic estimation under the minimum evolution principle. *BMC Evolutionary Biology*, **7**, 228+.

Cavalli-Sforza, L. L. and Edwards, A. W. F. (1967). Phylogenetic analysis. models and estimation procedures. *Am. J. Hum. Gen.*, **19**(3).

Charleston, M. A. (2001). Hitch-hiking : A parallel heuristic search strategy, applied to the phylogeny problem. *Journal of Computational Biology*, **8**(1), 79–91.

Cummings, M. P., Handley, S. A., Myers, D. S., Reed, D. L., Rokas, A., and Winka, K. (2003). Comparing bootstrap and posterior probability values in the four-taxon case. *Systematic biology*, **52**(4), 477–487.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, Albemarle Street, London.

De Jong, K. (1988). Learning with genetic algorithms : An overview. *Machine Learning*, **3**(2), 121–138.

Douady, C. J., Delsuc, F., Boucher, Y., Doolittle, W. F., and Douzery, E. J. (2003). Comparison of bayesian and maximum likelihood bootstrap measures of phylogenetic reliability. *Mol Biol Evol*, **20**(2), 248–254.

Elias, I. and Lagergren, J. (2008). Fast neighbor joining. *Theoretical Computer Science*.

Erixon, P., Svennblad, B., Britton, T., and Oxelman, B. (2003). Reliability of Bayesian Posterior Probabilities and Bootstrap Frequencies in Phylogenetics. *Syst Biol*, **52**(5), 665–673.

Felsenstein, J. (1985). Confidence limits on phylogenies : An approach using the bootstrap. *Evolution*, **39**(4), 783–791.

Felsenstein, J. (2004). *Inferring phylogenies*. Sunderland, MA : Sinauer Associates.

Felsenstein, J. (2005). Phylip (phylogeny inference package) version 3.6. Distributed by the author.

Fitch, W. and Margoliash, E. (1967a). A method for estimating the number of invariant amino acid coding positions in a gene using cytochrome c as a model case. *Biochemical Genetics*, **1**(1), 65–71.

Fitch, W. M. and Margoliash, E. (1967b). Construction of phylogenetic trees. *Science*, **155**(760), 279–284.

Friedman, N., Ninio, M., Pe'er, I., and Pupko, T. (2002). A structural em algorithm for phylogenetic inference. *J Comput Biol*, **9**(2), 331–353.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**(4), 711–732.

Guindon, S. and Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*, **52**(5), 696–704.

Hall, B. G. (2007). Evolveagene 3 : A dna coding sequence evolution simulation program : Nature precedings.

Hasegawa, M., Kishino, H., and Yano, T.-A. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of Molecular Evolution*, **22**(2), 160–174.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1), 97–109.

Hendy, M. D. and Penny, D. (1982). Branch and bound algorithm to determinate minimal evolutionary trees. *Math. Biosci.*, **60**, 309–368.

Hillis, D. M. (1996). Inferring complex phylogenies. *Nature*, **383**(6596), 130–131.

Huelsenbeck, J. and Rannala, B. (2004). Frequentist properties of bayesian posterior probabilities of phylogenetic trees under simple and complex substitution models. *Systematic Biology*, **53**(6), 904–913.

Huelsenbeck, J. P. and Ronquist, F. (2001). Mrbayes : Bayesian inference of phylogenetic trees. *Bioinformatics*, **17**(8), 754–755.

Huelsenbeck, J. P., Ronquist, F., Nielsen, R., and Bollback, J. P. (2001). Bayesian Inference of Phylogeny and Its Impact on Evolutionary Biology. *Science*, **294**(5550), 2310–2314.

Jones, G. (2008). On the reliability of bayesian posterior clade probabilities in phylogenetic analysis. *Nature Precedings*.

Jukes, T. H. and Cantor, C. R. (1969). *Evolution of Protein Molecules*. Academy Press.

Kimura, M. (1968). Evolutionary rate at the molecular level. *Nature*, **217**(5129), 624–626.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, **16**(2), 111–120.

Kimura, M. (1981). Estimation of evolutionary distances between homologous nucleotide sequences. *Proceedings of the National Academy of Sciences of the United States of America*, **78**(1), 454–458.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, **220**(4598), 671–680.

Kolaczkowski, B. and Thornton, J. W. (2007). Effects of branch length uncertainty on bayesian posterior probabilities for phylogenetic hypotheses. *Mol Biol Evol*, **24**(9), 2108–2118.

Kumar, S., Nei, M., Dudley, J., and Tamura, K. (2008). Mega : A biologist-centric software for evolutionary analysis of dna and protein sequences. *Briefings in bioinformatics*.

Lemmon, A. R. and Milinkovitch, M. C. (2002). The metapopulation genetic algorithm : An efficient solution for the problem of large phylogeny estimation. *Proc Natl Acad Sci U S A*, **99**(16), 10516–10521.

Lewis, P. O. (1998). A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Mol Biol Evol*, **15**(3), 277–283.

Lin, Y. (2008). *Tabu Search and Genetic Algorithm for Phylogeny Inference*. Ph.D. thesis, North Carolina State University.

Lundy, M. (1985). Applications of the annealing algorithm to combinatorial problems in statistics. *Biometrika*, **72**(1), 191–198.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, **21**(6), 1087–1092.

Michener, C. and Sokal, R. (1957). A quantitative approach to a problem in classification. *Evolution*, **11**, 130–162.

Mitchell, M. and Holland, J. H. (1993). When will a genetic algorithm outperform hill climbing ? In *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann, USA.

Posada, D. and Crandall, K. (1998). Modeltest : testing the model of dna substitution. *Bioinformatics*, **14**(9), 817–818.

Rambaut, A. and Grass, N. C. (1997). Seq-gen : an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**(3), 235–238.

Ronquist, F. and Huelsenbeck, J. P. (2003). Mrbayes 3 : Bayesian phylogenetic inference under mixed models. *Bioinformatics*, **19**(12), 1572–1574.

Saitou, N. and Nei, M. (1987). The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, **4**(4), 406–425.

Salter, L. A. and Pearl, D. K. (2001). Stochastic search strategy for estimation of maximum likelihood phylogenetic trees. *Systematic Biology*.

Schmidt, H. A. and von Haeseler, A. (2007). Maximum-likelihood analysis using tree-puzzle. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, **Chapter 6**.

Schmidt, H. A., Strimmer, K., Vingron, M., and von Haeseler, A. (2002). Tree-puzzle : maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*, **18**(3), 502–504.

Sheneman, Luke, Evans, Jason, Foster, and James, A. (2006). Clearcut : a fast implementation of relaxed neighbor joining. *Bioinformatics*, **22**(22), 2823–2824.

Simon, D. and Larget, B. (2000). Bayesian analysis in molecular biology and evolution (bambe).

Simonsen, M., Mailund, T., and Pedersen, C. N. (2008). Rapid neighbour-joining. In *WABI '08 : Proceedings of the 8th international workshop on Algorithms in Bioinformatics*, pages 113–122, Berlin, Heidelberg. Springer-Verlag.

Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, **28**, 1409–1438.

Stewart, C.-B. (2000). Lecture : Phylogenetic analysis.

Strimmer, K. and von Haeseler, A. (1996). Quartet Puzzling : A Quartet Maximum-Likelihood Method for Reconstructing Tree Topologies. *Mol Biol Evol*, **13**(7), 964–969.

Suzuki, Y., Glazko, G. V., and Nei, M. (2002). Overcredibility of molecular phylogenies obtained by bayesian phylogenetics. *PNAS*, **99**(25), 16138–16143.

Swofford, D. L. (2003). *PAUP*. Phylogenetic analysis using parsimony (*and other methods). Version 4*. Sinauer Associates, Sunderland, Massachusetts.

Tamura, K. and Nei, M. (1993). Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees. *Mol Biol Evol*, **10**(3), 512–526.

Tavaré, S. (1986). *Some probabilistic and statistical problems in the analysis of DNA sequences*. American Mathematical Society.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, **22**(4), 1701–1728.

Uzzell, T. and Corbin, K. W. (1971). Fitting discrete probability distributions to evolutionary events. *Science*, **172**(3988), 1089–1096.

Waddell, P. J. and Steel, M. A. (1997). General time-reversible distances with unequal rates across sites : Mixing [gamma] and inverse gaussian distributions with invariant sites. *Molecular Phylogenetics and Evolution*, **8**(3), 398 – 414.

Wehe, A., Bansal, M. S. S., Burleigh, J. G. G., and Eulenstein, O. (2008). Duptree : A program for large-scale phylogenetic analyses using gene tree parsimony. *Bioinformatics (Oxford, England)*.

Xia, X. and Xie, Z. (2001). Dambe : Software package for data analysis in molecular biology and evolution. *J Hered*, **92**(4), 371–373.

Yang, Z. (1993). Maximum-likelihood estimation of phylogeny from dna sequences when substitution rates differ over sites. *Mol Biol Evol*, **10**(6), 1396–1401.

Yang, Z. (1994). Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites : Approximate methods. *Journal of Molecular Evolution*, **39**(3), 306–314.

Yang, Z. and Rannala, B. (2005). Branch-length prior influences bayesian posterior probability of phylogeny. *Systematic Biology*, **54**(3), 455–470.

Zwickl, D. J. (2006). *Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion*. Ph.D. thesis, The University of Texas at Austin.